FPFIT, FPPLOT and FPPAGE:

Fortran computer programs for calculating and displaying earthquake fault-plane solutions

by

P. Reasenberg and D. Oppenheimer

U.S. Geological Survey
345 Middlefield Road
Menlo Park, California

Open-File Report No. 85-739

I. FPFIT Overview

Program FPFIT finds the double-couple fault-plane solution (source model) that best fits a given set of observed first motion polarities for an earthquake. The inversion is accomplished through a two-stage grid-search procedure that finds the source model minimizing a normalized, weighted sum of first-motion polarity discrepancies. Two weighting factors are incorporated in the minimization: one reflecting the estimated variance of the data, and one based on the absolute value of the theoretical P wave radiation amplitude (Aki and Richards, 1980). The latter weighting gives greater (lesser) weight to observations near radiation lobes (nodal planes). In addition to finding the minimum-misfit solution, FPFIT finds alternative solutions corresponding to significant relative minima in misfit. Such solutions, when they exist, generally correspond to faulting mechanisms distinctly different from the minimum-misfit solution, and may be deemed the preferred solution after consideration of possible data errors, unmodeled refractions and *a priori* knowledge of the tectonic environment. For each double-couple source model obtained, FPFIT formally estimates the uncertainty in the model parameters (strike, dip, rake). Finally, FPFIT calculates a uniformly distributed set of solutions within the range of estimated uncertainty. This set is used in the display program FPPLOT to graphically define the range of P-axis and T-axis orientation consistent with the data.

FPFIT calculates fault-plane solutions sequentially for a suite of earthquakes and accumulates statistics on the entire suite, such as the cumulative discrepancy rates for each station and for each data class (0-weight, 1-weight, etc..). FPFIT also accumulates statistics on the suite of misfit scores, estimated solution uncertainties, and other calculated quantities that characterize the overall quality of the inversion results. Based on an examination of these statistics, adjustments to the inversion procedure, such as revising the estimated data variance, or removing or reversing one or more individual seismograph stations, may be indicated. Such an iterative use of FPFIT provides an easy and sound way to identify and avoid some common problems in the computation of fault-plane solutions, including the presence of stations that consistantly receive refracted rays that cross a nodal plane, stations with reversed polarity, and noisy data.

Certain caution must be excercised when interpreting fault-plane solutions computed by FPFIT. Non-double-couple solutions are not considered in the program, and hence adoption of a source model obtained by FPFIT implies the assumption that the earthquake is a double-couple source. Furthermore, data errors, unmodeled refractions, and oversimplified layer boundaries in the hypocentral solution should always be critically assessed in the evaluation of fault-plane solutions. Hence, we strongly recommend that all applications of FPFIT include careful consideration of reported multiple solutions, as well as a visual check for unreported alternate solutions, and that the results not be uncritically adopted.

II. Computational Procedures used in FPFIT

For each earthquake, E^j, FPFIT compares the observed polarity at the k^{th} station with that calculated for a suite of source models $[M^i]$. A one-norm misfit function, $F^{i,j}$, is defined as

$$F^{i,j} = \frac{\sum_k \left\{ \left| p_o^{j,k} - p_t^{i,k} \right| \cdot W_o^{j,k} \cdot W_t^{i,k} \right\}}{\sum_k \left\{ W_o^{j,k} \cdot W_t^{i,k} \right\}} \qquad (1)$$

where $P_o^{j,k}$, $P_t^{i,k}$ are terms representing, respectively, the observed and theoretical first-motion polarity (0.5 for compression, -0.5 for dilatation). The term $W_o^{j,k}$ is the observation weight that must be estimated and assigned to the data. The term

$$W_t^{i,k} = \left[A(i,k)\right]^{1/2}$$

is the square root of the normalized theoretical P-wave radiation amplitude, A(i,k), expected at the k^{th} station for source model M^i. This weighting scheme down-weights observations near nodal planes, thereby minimizing the effect of inconsistancies near nodal planes, such as those caused by unmodeled refractions. The misfit function, $F^{i,j}$, is calculated for each source model in the suite, and the model that minimizes F is adopted as the fault-plane solution. An exceptional case occurs when more than one solution have identical values of $F^{i,j}$. This case may occur because, particularly in the case of models perfectly fitting the data, the misfit function may be flat-bottomed. To break the tie in such cases FPFIT applies the additional constraint that the denominator in Equation (1) be maximized. The effect of this constraint is to maximize the distance on the focal sphere between the observations and the nodal planes.

In computation each source model is represented by the strike, dip and rake of one of its nodal planes. Specification of one plane and the associated rake angle is, of course, sufficient to represent both the fault plane and the auxilliary plane. No special importance is attached to the particular plane used to represent each model, and representation of the final solution is consequently in terms of one plane and its rake. In the companion plotting programs FPPLOT and FPPAGE, both planes (and "P" and "T" axes) are calculated and displayed for each solution.

The procedure by which the suite of source models $[M^i]$ is tested is a two-stage three-dimensional grid search*. The first (coarse) stage uses 20° increments in each of the three parameters (strike, dip, and rake). All possible gridded values of rake and dip are included in the course search. However, only half the range of possible values of strike (from 0 to 160 degrees) is searched to avoid computing M^i for both the fault plane and its associated auxilliary plane. The course search identifies the solution corresponding to the minimum misfit, F_{min}, and, when they exist, multiple solutions corresponding to significant relative minima in misfit. The relative minima are detected in the course search up to a level of misfit $F \leq F_{min}$ + DFITC, where DFITC is an input parameter. Each of these solutions is then taken as the center of a second stage (fine) 3-dimensional grid search. The fine search uses grid point spacing of 5° for strike and dip, and 10° for rake. Parameter ranges in the fine search are, relative to the central value, ± 45° in strike and dip, ± 30° in rake. It is in the fine grid search that the final solutions are identified, and estimates of solution parameter uncertainty are calculated. When multiple solutions are identified for an event, the solutions are distinguished by an asterisk in the output files.

---

* In the following discussion we assume that the default values of the search parameters are used. The default values ensure that all regions in solution space are searched. When a restricted search (that is, one in which only a subset of the possible solutions is considered) is desired, the search parameters may be explicitly set in the input control file.

III. Estimation of Parameter Uncertainty and Solution Quality

It is possible to determine formal confidence intervals for the parameters of fault-plane solutions obtained with a grid-search procedure. The misfit measure (Equation 1) is a sum of terms that take on discrete values, most of which are zero. Accordingly, the variance in F, and the associated 90-percent one-sided confidence interval for F, are estimated from the data using the method described in Appendix A. Having obtained the confidence interval for F, we determine the range of each solution parameter, relative to the minimum-misfit value, within which the solution misfit is bounded by the confidence interval for F. These ranges are taken as the uncertainties for the solution parameters.

For each fault-plane solution, FPFIT calculates the uncertainties described above and several other quantitites designed to characterize the quality of the final solution. These quantities are reported, together with the fault-plane solution, in the program's output. They are:

1. F_j = minimum $[F_{i,j}]$, or a relative minimum of $F_{i,j}$. Note $F_j = 0.0$ represents a perfect fit to the data, while $F_j = 1.0$ represents a perfect misfit.

2. NOBS = number of observations used in the solution.

3. $\text{AVWT} = \frac{1}{\text{NOBS}} \sum_{k=1}^{\text{NOBS}} W_o^{j,k}$. AVWT is the mean data weight used in the solution; it is an overall measure of the quality of the data used in the solution. AVWT ranges from 0.0 to 30.0, with larger values reflecting

solutions computed from higher quality data.

4. $$\text{STDR} = \frac{\sum_{k=1}^{NOBS} \left(W_o^{j,k} \cdot W_t^{i,k} \right)}{\sum_{k=1}^{NOBS} W_o^{j,k}}$$. STDR is the station distribution ratio.

($0.0 \leq \text{STDR} \leq 1.0$). This quantity is sensitive to the distribution of the data on the focal sphere, relative to the radiation pattern. When this ratio has a low value (say, STDR < 0.5), then a relatively large number of the data lie near nodal planes in the solution. Such a solution is less robust than one for which STDR > 0.5, and, consequently, should be scrutinized closely and possibly rejected.

5. ΔSTR, ΔDIP, ΔRAK. These quantities are ranges (in degrees) of perturbations to the strike, dip and rake, respectively, of the final solution, that result in a misfit score bounded by the 90-percent confidence interval for F. ΔSTR, ΔDIP and ΔRAK are taken as the uncertainties in the solution parameters. When the distribution of the data on the focal sphere do not tightly constrain one or more parameters of the solution, the corresponding uncertainties are large; when the distribution of the data tightly constrain the solution these quantities are small.

FPFIT summarizes the quality of the adopted fault-plane solution with two letter codes, each of which may be "A", "B" or "C". The first letter code, QF, summarizes the value of F_j. The second quality code, QP, summarizes the three parameter uncertainties ΔSTR, ΔDIP, and ΔRAK, as follows:

F	QF
F $\leq$ 0.025	A
0.025 < F $\leq$ 0.1	B
F > 0.1	C

ΔSTR and ΔDIP and ΔRAK	QP
$\leq$ 20°	A
20° to 40°	B
> 40°	C

IV. Restricted Search Mode

FPFIT may be run in either of two modes. In the "unrestricted search" mode, all possible gridded solutions are tested. This mode is the usual mode of operation for FPFIT and is the one described in Section II. However, if the user chooses to search only a subset of the possible fault plane solutions, the "restricted search" mode may be used. In this mode the user may limit the ranges of strike, dip and rake from which test solutions are drawn in the search procedure. For example, the user may specify that only thrust solutions (rake = 90 degrees), or only vertical dextral strike-slip solutions (dip = 90 degrees and rake = 180 degrees) be considered. FPFIT will search for solutions only within the specified parameter ranges.

If the restricted search mode is used, two points should be kept in mind. First, the resulting solution may not be a true minimum-misfit (zero-derivative) solution, as the adopted solution may lie at the edge of one of the restricted parameter ranges. Second, the estimates of uncertainty for the solution parameters (ΔSTR, ΔDIP, ΔRAK, QF and QP are invalid, owing to the inability of FPFIT to perform its uncertainty analysis without a full range of solutions neighboring the adopted solution. Hence, the restricted search mode should only be used in special studies in which a "forced" solution is desired and estimates of solution uncertainty are not needed.

The unrestricted search mode is invoked by including in the input control file only lines 1, 2 and 3, and (optionally) station status lines, as in Appendix examples B.1.a and B.1.b (see also Section V.B).

The restricted search mode is invoked by including in the input control file an additional line specifying the ranges of the restricted search, as in Appendix examples B.1.c and B.1.d.

V. Input Files

A. Data File: FPFIT reads the print output file from program HYPO71 (Lee and Lahr, 1975). This file contains the hypocenter summary card, followed by (for each P-wave observation) the station-to-epicenter distance and azimuth, P-remark, angle of incidence, and flag denoting phase data discarded due to Jeffrey's weighting.

B. Control File (see appendix B.1): This file sets parameter values to tailor the computation to suit the particular requirements of the data set. At least three lines are required:

Line 1 - DISTMX, FMAGMN, MINOBS, IPRNT, IJEFF, NEV, DFITC (free format)

Name	Explanation
DISTMX	Maximum epicentral distance in km. Phase data from stations at epicentral distances greater than DISTMX are ignored.

FMAGMN — Minimum magnitude. Fault-plane solutions are not computed for events with magnitudes smaller than FMAGMN.

MINOBS — Minimum number of observations. Fault plane solutions are not computed for events for which NOBS < MINOBS.

IPRNT — Print output control. If IPRNT = 1, FPFIT generates a listing of the misfit function for each test solution in the fine search. Normally, IPRNT = 0.

IJEFF — Jeffrey's weighting control. When IJEFF = 1, inversion includes phase data discarded by Jeffrey's weighting in HYPO71. When IJEFF = 0, phase data discarded by Jeffrey's-weighting are omitted

NEV — Maximum number of events for which fault plane solutions will be calculated. Solutions are calculated sequentially from the input file until NEV events have been processed.

DFITC — Depth of search for relative minima in misfit. Relative minima with $F \leq F_{min}$ + DFITC are considered. We recommend the value DFITC=0.05 be used for most applications.

Line 2 - ERATE(1), ERATE(2), ERATE(3), ERATE(4) (free format)

Name	Explanation
ERATE	Estimated weighted error (discrepancy) rates for classes of hand-picked data. ERATE(J+1) corresponds to assigned pick quality J. Set ERATE(J+1) = 1.0 to exclude all data from class J. For 0.0 ≤ ERATE(J) < 0.5 lower values reflect better-quality data. Use actual weighted error rates reported in a previous run of FPFIT on same data for best estimates of ERATE.

Line 3 - ERATE(5), ERATE(6), ERATE(7), ERATE(8) (free format)

Name	Explanation
ERATE	Same as Line 2, except for classes of machine-picked (e.g., RTP) data, which are denoted by an 'X' in the first position of the P-remark field, as in XPUO. ERATE(J+5) corresponds to assigned pick quality J for machine-picked data.

Line 4, 5,..., N - STATUS, STATN (a1, 1x, a4) (Optional)

These lines are included if data from some stations are to be omitted from the inversion, or if first motion directions from some stations are to be reversed.

Name	Explanation
STATUS	If STATUS = 'R', FPFIT reverses first motion direction for all phase data from station 'STATN'. If STATUS = 'K', FPFIT ignores all phase data from station 'STATN'.
STATN	Station name

Line N+1 - Blank (Included only if next line is present)

Line N+2 - PHIOC, PHI1, DELOC, DEL1, XLAMOC, XLAM1, DPHIC, DELC, DLAMC, DPHIF, DDELF, DLAMF (free format) (Optional)

This line is included if a restricted search range or non-default values of the grid spacing are desired; if this line is omitted, the default search range values are used, as described in section IV.

Strike is measured clockwise from north; dip is measured down from horizontal; rake of 0 = left lateral, 90 = reverse, ± 180 = right lateral, -90 = normal.

Name	Explanation
PHIOC	Minimum value of coarse search strike range, in degrees ($0 \leq$ PHIOC ≤ 180)
PHI1	Maximum value of coarse search strike range, in degrees ($0 \leq$ PHI1 ≤ 180)
DELOC	Minimum value of coarse search dip range, in degrees ($0 \leq$ DELOC ≤ 90)
DEL1	Maximum value of coarse search dip range, in degrees ($0 \leq$ DEL1 ≤ 90)
XLAMOC	Minimum value of coarse search rake range, in degrees ($-180 \leq$ XLAMOC ≤ 180)
XLAM1	Maximum value of coarse search rake range, in degrees ($-180 \leq$ XLAM1 ≤ 180)
DPHIC	Strike increment, in degrees, for coarse search (0 < DPHIC)
DDELC	Dip increment, in degrees, for coarse search (0 < DDELC)
DLAMC	Rake increment, in degrees, for coarse search (0 < DLAMC)
DPHIF	Strike increment, in degrees, for fine search (0 < DPHIF)
DDELF	Dip increment, in degrees, for fine search (0 < DDELF)
DLAMF	Rake increment, in degrees, for fine search (0 < DLAMF)

VI. Output Files

A. Extended hypocenter summary card file (see Appendix B.2). For each earthquake satisfying the input criteria FMAGMN and MINOBS, FPFIT calculates a fault-plane solution and writes to this file an 'extended' summary card consisting of the original HYPO71 summary card concatenated with the fault-plane solution parameters. The format of the extended summary card is as follows:

Column	Explanation
1-80	HYPO71 summary card.
82-84	Dip direction (downdip azimuth in degrees, clockwise from north).
86-87	Dip angle in degrees down from horizontal.
88-91	Rake in degrees: 0=left lateral, 90=reverse, ±180=right lateral, -90=normal.
94-97	F_j
100-101	NOBS
103-107	AVWT
109-112	STDR
114-117	Ratio of number of machine-picked phases to total number of phases used in solution.
120-121	ΔSTR
123-124	ΔDIP
126-127	ΔRAK
129	Solution quality code QF
131	Solution quality code QP
132	Flag, normally blank. '*' indicates that solution is one of a set of multiple solutions found for this event.

B. Ray file (see Appendix B.3): This file is used as input to the plotting programs FPPLOT and FPPAGE. The first line in this file is the HYPO71 heading card. Following it is, for each solution: 1) the extended hypocenter summary card; 2) the number of additional fault-plane solutions in the suite of solutions corresponding to the 90-percent confidence interval for F_j; 3) the dip direction, dip angle and rake of each solution in the 90-percent confidence suite (Format 11(I4,I3,I4)); 4) the phase data used in the fault-plane determination and the normalized weights assigned to them.

The format of the phase cards is as follows:

Column	Explanation
2- 5	Station name.
7-11	Epicentral distance, in km.
13-17	Azimuth angle from epicenter to station (from north, in degrees).
19-23	Angle of incidence.
27-30	P-remark (eg., IPU2, XPD0, etc.)
32-35	Normalized observation weight W_0 used in inversion.
38	Discrepancy flag. If this field contains a '*', the observed polarity was discrepant with the adopted fault-plane solution. If the field is blank, observation was concordant.

C. Statistical summary file (see Appendix B.4): This file lists the options specified in the control file and used in the calculation, followed by an alphabetized summary listing of all stations used in the computation of all focal mechanisms. For each station, the summary reports the number of first motion polarity discrepancies, the number of agreements, the total number of observations, the weighted discrepancy rate (as in the numerator of Equation 1), and the fractional contribution from that station to the accumulated weighted error (misfit) from all stations for the suite of earthquakes processed. This last statistic is a measure of the effect of a station's discrepancy rate upon the entire inversion.

Following the station list are similar reports of the discrepancy rates for the hand-picked and machine-picked data, calculated separately for each data quality class (0, 1, 2, 3). The reported weighted error rate for each data class should be used as the input value for ERATE in a subsequent run of FPFIT with the same data. A double asterisk ('**') next to a reported weighted error rate indicates that this value differs from the corresponding (estimated) input value by more than 20 percent.

Next is a report of the distribution of the calculated fault-plane solution parameters F_j, ΔDIP, ΔSTR, and ΔRAK. These distributions provide a measure of the variance in the fault-plane set, and highlight the presence of solutions with unusually large uncertainties.

D. Fit file (see Appendix B.5): This file is optionally generated (when

IPRNT = 1). For each solution the extended summary card is followed by a listing of the 3-dimensional fit parameter matrix. This matrix contains the misfit scores $F_{i,j}$ (multiplied by 1000) for the source models calculated in the fine grid search, and is organized according to strike, dip and rake. The misfit score for the adopted solution is annotated with an "A"; the "additional solutions", corresponding to the 90-percent confidence interval for F, are annotated by an "*".

Examination of this matrix shows the behavior of $F_{i,j}$ for solutions near the adopted solution and provides some insight as to whether the adopted solution is well-constrained. This listing file consists of 3 printer pages per fault-plane solution for the default grid search range.

VII. FPFIT Installation Considerations

A. The dimension of all arrays are variable, and are set through assignments in the PARAMETER statement at the beginning of the main program. Thus, the array dimensions can be easily modified by changing the variable assignments in the PARAMETER statement and recompiling.

B. All I/O statements refer to variable names for logical unit numbers. These variables are also set through assignment in the PARAMETER statement. In this way, the program can be tailored to use any convenient logical unit numbers by changing the assignments in the PARAMETER statement and recompiling.

C. This program was developed on a DEC VAX 11/780 computer with a VMS operating system. The FORTRAN compiler has an extension to the FORTRAN 77 standard which permits character strings following an exclamation point (!) anywhere on a line to be interpreted as a comment. FPFIT takes advantage of this extension to describe the function of each variable in the code on the same line as the variable declaration. These embedded comments may need to be removed for other compilers.

D. The OPEN statement for logical unit SUNIT in the main program contains the non-standard argument 'CARRIAGECONTROL=LIST'. Under VMS this changes the attributes of the file such that the printer does not interpret the first column of each line as a print control character. Similarly, all WRITE statements to logical unit SUNIT do not have carriage control characters in the first column. The OPEN statement for logical units IUNIT and CUNIT contain the non-standard argument "READONLY". Under VMS this permits the program to open files for which the user does not have "Write Status". This argument may have to be removed for use with other operating systems.

VIII. FPPLOT Overview

FPPLOT is an interactive plotting program for displaying fault plane solutions calculated by FPFIT. FPPLOT produces one frame of graphic output for each solution found by FPFIT. The input file for FPPLOT is the "RAY" output file produced by FPFIT (see section VI.B). What follows is a brief description of the graphic output from FPPLOT. Refer to any of the examples in Appendix B.6, all of which were produced with FPPLOT.

The top line is an image of the "Heading Card" optionally included in the HYPO71 input file used to locate the earthquake. It is reproduced here to provide a convenient label that associates the fault-plane solution with the hypocenter location model.

Below the heading card is the extended summary card, on two lines, as described in Section VI.A.

The 5.75-inch-diameter circle below the extended summary card is a lower-hemisphere equal-area projection of the adopted fault-plane solution and first-motion data. Compressional rays are represented by circle symbols, dilational rays by triangle symbols. Upgoing (direct) rays are indicated by bold-face symbols, downgoing (refracted) rays by light-lined symbols. The size of the symbol is proportional to the observational weight W_0 associated with the ray. The ray symbols are centered on the points of the projection to which they correspond. The bold-face symbols "P" and "T" are centered on the points corresponding to the "pressure-axis" and "tension-axis", respectively, for the adopted solution.

To the right of the 5.75-inch-diameter circle is a table listing the observations that are discrepant with the adopted fault-plane solution. The table includes, from left to right for each discrepant observation, the station name, epicentral distance, azimuth in degrees from north, angle of incidence in degrees, and the P-remark. If no observations are discrepant, this table is omitted.

The 2.5-inch-diameter circle in the lower right corner of the page is a lower-hemisphere equal-area projection showing the position of the P-axis and T-axis corresponding to the adopted fault-plane solution (bold-face "P" and "T" symbols, respectively). A set of additional pairs of P- and T-axes are plotted (light lines), corresponding to a set of solutions with misfit scores within the 90-percent confidence interval for F. The resulting distribution of P- and T-axes represents the range of orientations of P and T consistent with the data, allowing for uncertainty.

IX. FPPLOT and FPPAGE Installation Considerations

In addition to the installation considerations for FPFIT (see section VI), the following apply:

A. VMS requires that character variables be passed to plot calls by "reference". Therefore, character variables in subroutine argument lists are themselves arguments of the VMS system function %REF. This function may be omitted for use with other operating systems.

B. The subroutine DELAY calls VMS system routines to achieve a half second delay following the clearing of the plot screen for Tektronix plotting terminals. Installation of PLOTFM on other operating systems will require the corresponding system calls.

C. All calls to graphics rountines conform to standard Versatec or Calcomp software.

X. FPPAGE Overview

FPPAGE is an interactive plotting program for displaying on a single page up to 42 fault plane solutions calculated by FPFIT. The input file for FPPAGE is the "RAY" output file produced by FPFIT (see section VI.B). Refer to the example in Appendix B.7.

Each fault plane solution is represented by a lower-hemisphere equal-area projection. Above each projection is plotted a header consisting, according to the user request, of either the sequential number of the earthquake in the "RAY" file, or the origin time of the earthquake. The header is annotated with an asterisk (*) to indicate multiple solutions. Compressional rays are depicted as solid circles; dilational rays as open circles. Plotting of the first motion symbols may be suppressed by the user. Finally the P- and T-axes of the solution are plotted. If first-motion plotting is suppressed, only the T-axis is plotted.

XI. References

Aki, K., and P. G. Richards, 1980, Quantitative Seismology, Theory and Methods, Vol. 1, W. H. Freeman and Co., San Francisco, California.

Lee, W. H. K., and J. C. Lahr, 1975, HYPO71 (Revised): A computer program for determining hypocenter, magnitude, and first motion pattern of local earthquakes, U.S. Geol. Surv. Open File Rep., 75-311, 114 pp.

ACKNOWLEDGEMENTS

Some of the computer routines used in FPFIT, FPPAGE and FPPLOT were written by our colleagues at the U.S.G.S. in Menlo Park. John Lahr and Fred Klein kindly provided some of the graphics routines; Bruce Julian provided the moment tensor computation routines and showed us how to increase the computational speed of FPFIT with their use. Mark Mathews, at Stanford University, assisted us with the error analysis used in determining the uncertainties in the solution parameters. Donna Eberhart-Phillips assisted in the development of an early unpublished version of FPFIT.

We have had encouraging and stimulating discussions with many colleagues that were helpful in shaping these programs. We especially appreciate the comments of Jerry Eaton, Bruce Julian, John Lahr, Al Lindh, and Chris Stephens. We thank Bruce Julian and John Lahr for their reviews of the manuscript.

APPENDIX A

We wish to know the variance of F in Equation (1) solely from *a priori* knowledge of the variance of the data. To do this we take advantage of the fact that the data are provided with assigned "quality codes" (0, 1, etc.) that we assume correspond to uniform variances for each data class. This allows us to treat F as a sum of binomial processes. For one earthquake, let

n = the number of observations

N = the number of data classes represented by the observations

n_j = the number of observations in data class j.

$$n = \sum_{j=1}^{N} n_j$$

Now let

p_k = the error (discrepancy) probability for the k'th observation

r_j = the error (discrepancy) rate for class j.

We may simplify Equation (1) by combining the weighting terms W_o^k and W_t^k into a single term

$$W_k = W_o^k \cdot W_t^k$$

and expressing the misfit term

$$m_k = \left| P_o^k - P_t^k \right|$$

where

$$m_k = \begin{cases} 1, & \text{when the k'th observation is a misfit;} \\ 0, & \text{otherwise.} \end{cases}$$

Then

$$F = \frac{\sum_{k=1}^{n} \left(m_k \cdot W_k \right)}{\sum_{k=1}^{n} W_k} \tag{A.1}$$

The appropriate weighting terms in (A.1) are

$$W_k = \frac{1}{\sigma_k} = \frac{1}{\sqrt{p_k(1-p_k)}}$$

Equation (A.1) can be rewritten in terms of the observations and weights in each data class as

$$F = \frac{\sum_{j=1}^{N} \sum_{k=1}^{n_j} \left(m_{j_k} \cdot W_{j_k} \right)}{\sum_{j=1}^{N} \sum_{k=1}^{n_j} W_{j_k}}$$

$$= \frac{\sum_{j=1}^{N} \tilde{W}_j \sum_{k=1}^{n_j} m_{j_k}}{\sum_{j=1}^{N} \sum_{k=1}^{n_j} \tilde{W}_j}$$

where $\tilde{W}_j = \dfrac{1}{\sigma_j} = \dfrac{1}{\sqrt{r_j(1-r_j)}}$.

Finally,

$$F = \frac{\sum_{j=1}^{N} \left(\tilde{m}_j \cdot \tilde{W}_j \right)}{\sum_{j=1}^{N} \left(n_j \cdot \tilde{W}_j \right)} \tag{A.2}$$

where

$$\tilde{m}_j = \sum_{k=1}^{n_j} m_k$$

is the number of discrepancies in data class j. Equation (A.2) is an expression for F in terms of the number of discrepancies in each data class and the (uniform) weights assigned to observations in each data class. The variance of F is easily obtained from (A.2):

$$\text{Var}(F) = \frac{\sum_{j=1}^{N} \left(\text{Var}(\tilde{m}_j) \cdot \tilde{W}_j^{\ 2}\right)}{\left(\sum_{j=1}^{N} n_j \cdot \tilde{W}_j\right)^2}$$

$$\text{Var}(\tilde{m}_j) = n_j \, r_j \, (1-r_j) = n_j/\tilde{W}_j^{\ 2}$$

$$\text{Var}(F) = \frac{\sum_{j=1}^{N} n_j}{\left(\sum_{j=1}^{N} n_j \cdot \tilde{W}_j\right)^2} = \frac{n}{\left(\sum_{j=1}^{N} n_j \cdot \tilde{W}_j\right)^2} . \qquad \text{(A.3)}$$

Finally, the standard deviation of F

$$\sigma_F = \sqrt{\text{Var}(F)}$$

is calculated. The 90-percent one-sided confidence interval for F is estimated from σ_F by assuming that F is normally distributed.

To implement this estimation procedure we must start with some estimates of the terms $[r_j]$, recalling that these discrepancy rates include contributions from both data errors and modeling errors. While we may have some idea of the contribution to $[r_j]$ from data (reading) errors, we cannot know the contribution from modeling (refraction) errors until after the inversion is done. Therefore the first run of FPFIT requires an educated guess for the [r_j]. Upon completion of the inversion for a suite of earthquakes, FPFIT reports the actual accumulated data-class error rates [r_j] for the entire suite. These rates ("WEIGHTED ERROR RATE") should be used as initial estimates in the subsequent run of FPFIT. Such an iterative use of FPFIT provides a "bootstrap" ability to estimate formal error confidence limits from the data.

APPENDIX B

B.1 Examples of CONTROL FILEs for processing a set of fault-plane solutions.

B.2 Example of EXTENDED SUMMARY CARD FILE.

B.3 A portion of the RAY FILE corresponding to one selected earthquake.

B.4 STATISTICAL SUMMARY FILE resulting from a run of FPFIT on a suite of 40 earthquakes.

B.5 A portion of the FIT FILE corresponding to one selected earthquake.

B.6 Graphic output from program FPPLOT for selected earthquakes.

B.7 Graphic output from program FPPAGE.

B.1 Examples of CONTROL FILEs for FPFIT.

a) Control file corresponding to the statistical summary file shown in B.4.

b) Control file that excludes all machine-picked data.

c) Control file that allows only vertical right-lateral slip fault planes.

d) Control file that allows only pure thrust solutions.

a)

```
999. Ø.Ø 15 Ø Ø 999 Ø.Ø5
Ø.Ø4 Ø.Ø6 Ø.1Ø Ø.12
Ø.15 1.ØØ 1.ØØ 1.ØØ
R FRI
R JAS
R KPK
R MGL
R MIN
R MINB
R ORV
R  FRI
R  JAS
R  KPK
R  MGL
R  MIN
R  ORV
K ABG
K DEF
K GHI
```

b)

```
999. Ø.Ø 15 Ø Ø 99 Ø.Ø5
Ø.Ø7 Ø.Ø9 Ø.1Ø Ø.13
1.ØØ 1.ØØ 1.ØØ 1.ØØ
```

c)

```
999. Ø.Ø 15 Ø Ø 99 Ø.Ø5
Ø.Ø7 Ø.Ø9 Ø.1Ø Ø.13
1.ØØ 1.ØØ 1.ØØ 1.ØØ

Ø. 16Ø. 9Ø. 9Ø. 18Ø. 18Ø. 2Ø. 2Ø. 2Ø. 5. 5. 1Ø.
```

d)

```
999. Ø.Ø 15 Ø Ø 99 Ø.Ø5
Ø.Ø7 Ø.Ø9 Ø.1Ø Ø.13
1.ØØ 1.ØØ 1.ØØ 1.ØØ

Ø. 16Ø. 1Ø. 9Ø. 9Ø. 9Ø. 2Ø. 2Ø. 2Ø. 5. 5. 1Ø.
```

B.2 EXTENDED SUMMARY CARD FILE

841211	759	14.60	38-49.07	122-49.71	0.12	1.77	28	38	3.0	0.19	0.4	0.7	B	50	60	170	0.02	21	4.24	0.65	0.19	10	45	30	AIB
841225	1218	36.29	38-49.43	122-47.05	0.16	1.39	21	40	2.0	0.07	0.2	0.3	A	315	85	40	0.09	19	4.15	0.51	0.16	10	15	10	BIA
850910	647	26.13	38-49.03	122-49.44	0.28	1.27	18	45	3.0	0.18	0.5	0.8	B	145	80	30	0.03	17	4.48	0.61	0.00	10	35	30	BIB
841123	17 6	19.33	38-48.65	122-48.66	0.64	2.02	30	33	2.0	0.17	0.4	0.6	B	145	90	30	0.02	24	3.99	0.67	0.33	15	20	40	AIB
850907	1052	5.47	38-48.98	122-48.81	0.67	1.45	25	34	2.0	0.09	0.2	0.3	A	215	35	100	0.05	22	4.61	0.50	0.00	25	5	20	BIB
850306	1411	50.10	38-47.55	122-46.61	0.68	1.09	17	52	2.0	0.18	0.5	0.6	B	200	40	80	0.05	17	4.79	0.48	0.00	20	5	30	BIB
850504	1622	16.78	38-51.41	122-51.25	0.75	1.41	19	60	8.0	0.09	0.3	0.4	B	130	45	-70	0.06	16	4.38	0.57	0.00	15	10	30	BIB
850728	2010	31.85	38-47.08	122-46.02	0.84	1.57	23	38	2.0	0.09	0.2	0.2	A	165	35	20	0.00	18	4.30	0.63	0.00	15	30	30	AIB
850808	19 7	5.89	38-47.53	122-46.38	0.87	1.77	24	48	2.0	0.07	0.2	0.2	A	155	80	10	0.04	19	4.59	0.63	0.00	10	25	30	BIB
850925	437	30.48	38-46.82	122-45.97	0.91	2.04	27	37	2.0	0.08	0.2	0.2	A	325	80	0	0.08	22	4.54	0.63	0.00	10	40	30	BIB
851006	1030	22.32	38-49.35	122-46.48	0.95	1.74	25	45	2.0	0.07	0.2	0.2	A	180	80	-40	0.01	20	4.66	0.62	0.00	20	50	30	AIB
850509	314	18.95	38-49.43	122-48.05	0.96	1.75	24	40	2.0	0.07	0.2	0.2	A	115	75	-30	0.04	23	4.29	0.62	0.00	10	20	20	BIA
840404	19 0	34.59	38-48.66	122-48.01	0.97	2.04	20	39	1.0	0.06	0.2	0.1	A	70	80	160	0.09	20	4.50	0.63	0.00	5	30	40	BIB
841220	154	50.27	38-49.27	122-49.47	0.99	1.72	22	46	3.0	0.19	0.5	0.6	B	95	55	-130	0.05	19	4.03	0.57	0.16	25	15	30	BIB
850529	010	37.58	38-48.08	122-48.70	1.02	1.80	24	46	2.0	0.07	0.2	0.2	A	230	85	170	0.00	18	4.56	0.64	0.00	10	30	10	AIB
850729	2157	12.27	38-47.90	122-46.93	1.03	1.67	24	31	1.0	0.07	0.1	0.2	A	190	70	-30	0.02	18	4.32	0.65	0.00	15	25	40	AIB
850708	1856	46.64	38-48.02	122-46.82	1.06	1.48	23	35	1.0	0.10	0.2	0.2	A	145	90	-30	0.00	16	4.55	0.63	0.00	10	20	40	AIB
850415	2150	27.52	38-48.04	122-48.81	1.06	1.85	27	35	2.0	0.08	0.2	0.2	A	60	70	-150	0.01	22	4.57	0.57	0.00	10	15	20	AIA
850415	2150	27.52	38-48.04	122-48.81	1.06	1.85	27	35	2.0	0.08	0.2	0.2	A	80	50	-120	0.04	22	4.57	0.49	0.00	10	5	10	BIA*
841222	1537	43.29	38-47.65	122-46.17	1.12	1.34	18	43	2.0	0.04	0.1	0.1	A	320	75	30	0.04	16	4.43	0.53	0.13	5	25	60	BIB
840617	624	51.41	38-47.25	122-46.34	1.18	1.52	19	37	2.0	0.04	0.1	0.1	A	140	65	-30	0.09	18	4.14	0.61	0.00	20	35	30	BIB
850914	13 3	26.97	38-48.73	122-47.44	1.21	1.39	23	40	1.0	0.07	0.2	0.2	A	90	50	-100	0.10	18	4.76	0.48	0.00	25	10	20	CIB
850727	1059	16.89	38-48.81	122-48.52	1.22		23	44	1.0	0.08	0.2	0.2	A	205	85	170	0.04	19	4.45	0.61	0.00	20	25	20	BIB
850831	1611	7.68	38-48.69	122-48.12	1.28	1.76	27	38	1.0	0.07	0.2	0.2	A	80	70	-120	0.08	19	4.47	0.58	0.00	25	15	30	BIB
841103	1733	3.16	38-47.44	122-46.41	1.33	2.22	17	54	2.0	0.04	0.1	0.1	A	220	75	170	0.02	17	4.84	0.59	0.00	15	30	20	AIB
841221	1128	55.00	38-48.65	122-47.43	1.34	1.96	27	39	0.0	0.14	0.3	0.3	A	215	65	150	0.08	26	3.94	0.59	0.31	15	25	30	BIB
850905	2043	38.09	38-48.78	122-49.14	1.34	1.34	19	44	2.0	0.07	0.2	0.3	A	240	45	90	0.02	17	4.58	0.45	0.00	15	5	10	AIA
841222	9 0	41.52	38-49.65	122-49.13	1.34	1.40	21	46	2.0	0,09	0.2	0.3	A	145	85	-30	0.04	17	4.44	0.62	0.18	10	15	20	BIA
850403	839	35.64	38-47.79	122-46.71	1.36	2.65	21	32	2.0	0.03	0.1	0.1	A	235	80	170	0.00	22	4.86	0.64	0.00	10	25	10	AIB
850907	2226	5.80	38-47.26	122-46.55	1.37	1.44	22	38	2.0	0.04	0.1	0.1	A	200	75	-20	0.05	16	4.66	0.62	0.00	10	30	30	BIB
850907	2226	5.80	38-47.26	122-46.55	1.37	1.44	22	38	2.0	0.04	0.1	0.1	A	190	55	-30	0.06	16	4.66	0.55	0.00	10	20	10	BIA*
850828	112	10.05	38-48.40	122-47.56	1.37	1.43	20	38	0.0	0.08	0.2	0.2	A	220	80	160	0.00	17	4.74	0.61	0.00	10	20	20	AIA
840410	4 2	2.49	38-48.52	122-47.63	1.38	2.95	25	38	0.0	0.05	0.1	0.1	A	75	85	-140	0.02	25	4.09	0.56	0.00	10	15	20	AIA
850810	1934	16.86	38-49.53	122-47.81	1.38	1.20	20	40	1.0	0.08	0.2	0.2	A	140	35	-150	0.09	16	4.42	0.50	0.00	25	5	20	BIB
850810	1934	16.86	38-49.53	122-47.81	1.38	1.20	20	40	1.0	0.08	0.2	0.2	A	110	65	-130	0.12	16	4.42	0.60	0.00	15	20	10	CIA*
850829	15 4	9.01	38-47.02	122-46.15	1.42	1.74	17	38	2.0	0.04	0.1	0.1	A	220	70	150	0.04	16	4.38	0.55	0.00	5	10	20	BIA
841200	440	48.59	38-48.23	122-47.10	1.43	2.46	20	44	1.0	0.05	0.1	0.1	A	350	85	10	0.00	20	5.01	0.61	0.00	5	25	20	AIB
850410	610	25.03	38-49.59	122-48.39	1.47	1.56	24	41	1.0	0.05	0.1	0.1	A	100	50	-140	0.00	17	4.57	0.56	0.00	10	15	30	AIB
841222	1633	41.77	38-47.41	122-46.46	1.52	2.71	27	46	2.0	0.09	0.2	0.5	A	245	70	-180	0.02	31	3.93	0.62	0.00	10	35	20	AIB
850731	350	43.96	38-48.35	122-50.39	1.53	1.32	22	45	4.0	0.07	0.2	0.5	A	160	55	-100	0.01	18	4.56	0.45	0.00	10	15	10	AIA
850615	037	32.10	38-49.42	122-48.42	1.54	1.48	19	46	2.0	0.05	0.1	0.3	A	330	70	-10	0.00	16	4.55	0.70	0.00	10	45	40	AIB
841213	1410	13.86	38-48.98	122-48.66	1.55	1.55	18	72	2.0	0.07	0.2	0.4	A	105	35	-140	0.03	18	4.48	0.59	0.00	15	20	30	BIB
850719	1549	14.24	38-49.52	122-46.92	1.55	2.40	22	41	2.0	0.04	0.1	0.2	A	255	80	160	0.00	23	4.51	0.70	0.00	15	50	30	AIB
850511	1428	18.14	38-48.05	122-47.88	1.57	2.35	20	36	1.0	0.05	0.1	0.3	A	35	80	10	0.00	20	4.39	0.81	0.00	15	55	40	AIB
850606	1012	17.71	38-48.03	122-47.02	1.59	1.35	19	56	1.0	0.04	0.1	0.2	A	235	75	170	0.00	17	4.36	0.70	0.00	10	45	20	AIB
850706	357	33.00	38-49.40	122-49.02	1.61	1.85	22	45	2.0	0.04	0.1	0.3	A	100	60	-100	0.05	19	4.45	0.55	0.00	15	35	10	BIB
850601	1750	11.19	38-49.09	122-49.17	1.61	1.94	20	83	3.0	0.04	0.1	0.3	A	70	70	-100	0.07	17	4.79	0.47	0.00	45	10	20	BIB
850216	2144	15.48	38-46.97	122-45.64	1.61	2.19	29	47	2.0	0.07	0.2	0.3	A	240	75	160	0.00	24	4.33	0.64	0.00	5	35	30	AIB
850410	16 7	39.14	38-47.19	122-45.53	1.65	1.70	17	112	2.0	0.05	0.2	0.3	B	170	85	-50	0.03	16	4.52	0.55	0.00	10	10	20	BIA
841220	155	22.60	38-49.23	122-49.17	1.71	2.47	20	33	3.0	0.09	0.2	0.5	A	85	65	-120	0.00	34	4.09	0.56	0.29	15	10	20	BIA
850405	934	23.85	38-48.13	122-48.37	1.72	2.15	30	63	1.0	0.06	0.1	0.2	A	330	85	40	0.05	28	4.23	0.58	0.00	15	20	40	BIB
850412	2041	34.62	38-48.20	122-48.29	1.73	2.17	23	40	1.0	0.05	0.1	0.2	A	60	80	-130	0.05	23	4.63	0.55	0.00	10	10	20	BIA
850603	2049	27.66	38-49.11	122-48.69	1.73	1.77	24	42	2.0	0.08	0.2	0.4	A	160	30	-30	0.05	19	4.33	0.59	0.00	25	25	20	BIB
850531	613	39.22	38-48.06	122-48.69	1.76	2.11	30	32	2.0	0.06	0.1	0.3	A	130	80	10	0.00	22	4.62	0.81	0.00	15	45	50	AIB
850909	1319	6.34	38-49.23	122-49.35	1.78	1.51	20	45	3.0	0.09	0.3	0.6	A	145	90	80	0.07	16	4.55	0.43	0.00	15	20	20	BIA
850909	1319	6.34	38-49.23	122-49.35	1.78	1.51	20	45	3.0	0.09	0.3	0.6	A	80	10	-150	0.08	16	4.55	0.45	0.00	15	30	10	BIB*
850419	1838	17.68	38-49.26	122-48.67	1.80	2.00	22	44	2.0	0.06	0.2	0.3	A	95	75	-90	0.05	20	4.56	0.50	0.00	30	10	10	BIB

B.3 Portion of the RAY FILE for one selected earthquake.

```
85Ø5Ø9  314 18.96 38-49.43 122-48.Ø5   Ø.96   1.75 24  4Ø  2.Ø Ø.Ø7  Ø.2  Ø.2 A  115 75 -3Ø  Ø.Ø4  23  4.29 Ø.62 Ø.ØØ  1Ø 2Ø 2Ø BIA
        9Ø
  9Ø 5Ø -7Ø  95 55 -6Ø 1Ø5 6Ø -5Ø 11Ø 65 -5Ø 115 65 -5Ø 11Ø 7Ø -5Ø 115 7Ø -5Ø 12Ø 7Ø -5Ø 1ØØ 55 -4Ø 1ØØ 6Ø -4Ø 1Ø5 6Ø -4Ø
 11Ø 6Ø -4Ø 1ØØ 65 -4Ø 1Ø5 65 -4Ø 11Ø 65 -4Ø 115 65 -4Ø 1Ø5 7Ø -4Ø 11Ø 7Ø -4Ø 115 7Ø -4Ø 12Ø 7Ø -4Ø 1Ø5 75 -4Ø 11Ø 75 -4Ø
 115 75 -4Ø 12Ø 75 -4Ø 11Ø 8Ø -4Ø 115 8Ø -4Ø 12Ø 8Ø -4Ø 125 8Ø -4Ø 115 85 -4Ø 12Ø 85 -4Ø 125 85 -4Ø 12Ø 9Ø -4Ø 125 9Ø -4Ø
 1ØØ 55 -3Ø 1Ø5 55 -3Ø 1Ø5 6Ø -3Ø 11Ø 6Ø -3Ø 1Ø5 65 -3Ø 11Ø 65 -3Ø 115 65 -3Ø 12Ø 65 -3Ø 1Ø5 7Ø -3Ø 11Ø 7Ø -3Ø 115 7Ø -3Ø
 12Ø 7Ø -3Ø 1Ø5 75 -3Ø 11Ø 75 -3Ø 12Ø 75 -3Ø 1Ø5 8Ø -3Ø 11Ø 8Ø -3Ø 115 8Ø -3Ø 12Ø 8Ø -3Ø 125 8Ø -3Ø 1Ø5 85 -3Ø 11Ø 85 -3Ø
 115 85 -3Ø 12Ø 85 -3Ø 125 85 -3Ø 11Ø 9Ø -3Ø 115 9Ø -3Ø 12Ø 9Ø -3Ø 125 9Ø -3Ø 295 85  3Ø 3ØØ 85  3Ø 3Ø5 85  3Ø 11Ø 6Ø -2Ø
 11Ø 65 -2Ø 115 65 -2Ø 12Ø 65 -2Ø 11Ø 7Ø -2Ø 115 7Ø -2Ø 12Ø 7Ø -2Ø 11Ø 75 -2Ø 115 75 -2Ø 12Ø 75 -2Ø 11Ø 8Ø -2Ø 115 8Ø -2Ø
 12Ø 8Ø -2Ø 125 8Ø -2Ø 115 85 -2Ø 12Ø 85 -2Ø 115 9Ø -2Ø 12Ø 9Ø -2Ø 115 65 -1Ø 11Ø 7Ø -1Ø 115 7Ø -1Ø 12Ø 7Ø -1Ø 115 75 -1Ø
 12Ø 75 -1Ø 115 8Ø -1Ø
GMM     1.6   6.Ø 121.Ø   IPDØ 1.ØØ
GDX     1.9 161.Ø 117.Ø   IPDØ 1.ØØ
GCM     4.4 117.Ø 1Ø2.Ø   IPDØ 1.ØØ
GSM     6.3 164.Ø  99.Ø   IPDØ 1.ØØ
GGPV    7.6 21Ø.Ø  6Ø.Ø   IPUØ 1.ØØ
GGL     8.4  15.Ø  6Ø.Ø   IPDØ 1.ØØ
GCR     9.2 127.Ø  6Ø.Ø   IPUØ 1.ØØ
GSG     9.2  59.Ø  6Ø.Ø   IPUØ 1.ØØ
GBG    1Ø.6  96.Ø  6Ø.Ø   IPUØ 1.ØØ
GPM    12.9 282.Ø  6Ø.Ø   IPUØ 1.ØØ
GAX    13.2 162.Ø  6Ø.Ø   EPD2 Ø.65
GMK    16.2   4.Ø  6Ø.Ø   IPDØ 1.ØØ
GRT    17.1  42.Ø  6Ø.Ø   EPU2 Ø.65
GCV    19.6 252.Ø  6Ø.Ø   EPU2 Ø.65
NMH    22.6 139.Ø  54.Ø   EPU2 Ø.65
GSS    22.9 234.Ø  54.Ø   EPU2 Ø.65
NHB    27.7 2ØØ.Ø  54.Ø   IPU1 Ø.83
GMC    28.8 263.Ø  54.Ø   IPU1 Ø.83
GHL    3Ø.6 322.Ø  54.Ø   EPD2 Ø.65
NMT    3Ø.9  94.Ø  54.Ø   IPU1 Ø.83
GMO    32.3 247.Ø  54.Ø   EPD2 Ø.65
GHG    33.9 357.Ø  54.Ø   EPD2 Ø.65
GDC    38.5 261.Ø  53.Ø   EPU2 Ø.65
```

B.4 STATISTICAL SUMMARY FILE

```
 VELEST MODEL LGAR27 WITH REMAINING CALNET *'D
CONTROL FILE = GEYS.INP;16
MAXIMUM EPICENTRAL DISTANCE =    999.0000
MINIMUM MAGNITUDE =   0.00000000E+00
MINIMUM # OBSERVATIONS =           15
PARAMETER FIT FILE SUPPRESSED (IPRNT = 0)
PHASE DATA REJECTED BY JEFFREYS' WEIGHTING EXCLUDED (IJEFF = 0)
UPTO          999  EVENTS PROCESSED
MISFIT RANGE FOR RELATIVE MINIMA IN COARSE SEARCH =   5.0000001E-02

HAND-PICKED DATA:     QUALITY     WEIGHT     WEIGHTED ERROR RATE (EST.)
                         0          5.103      0.040
                         1          4.211      0.060
                         2          3.333      0.100
                         3          3.077      0.120

MACHINE-PICKED DATA: QUALITY     WEIGHT     WEIGHTED ERROR RATE (EST.)
                         0          2.801      0.150
                         1          0.000      1.000
                         2          0.000      1.000
                         3          0.000      1.000

UNRESTRICTED SEARCH RANGE:          COARSE SEARCH                          FINE SEARCH
                         START      END    INCRMNT  # INCRMNTS        INCRMNT  # INCRMNTS
                         ----------------------------------        ------------------
                STRIKE:    0.0    160.0    20.0           9           5.0         19
                   DIP:   10.0     90.0    20.0           5           5.0         19
                  RAKE: -180.0    160.0    20.0          18          10.0          7

***** EVENT 850415 2150 27.52 HAS MULTIPLE SOLUTIONS  *****
***** EVENT 850907 2226  5.80 HAS MULTIPLE SOLUTIONS  *****
***** EVENT 850810 1934 16.86 HAS MULTIPLE SOLUTIONS  *****
***** EVENT 850909 1319  6.34 HAS MULTIPLE SOLUTIONS  *****
***** EVENT 841217 1334 58.86 HAS MULTIPLE SOLUTIONS  *****
***** EVENT 840516 1236 55.36 HAS MULTIPLE SOLUTIONS  *****
***** EVENT 841210  6 9  0.59 HAS MULTIPLE SOLUTIONS  *****
***** EVENT 850130  6 4  9.26 HAS MULTIPLE SOLUTIONS  *****
***** EVENT 850815 2047 23.99 HAS MULTIPLE SOLUTIONS  *****
***** EVENT 850701 1430 31.23 HAS MULTIPLE SOLUTIONS  *****
***** EVENT 850905 1324 19.56 HAS MULTIPLE SOLUTIONS  *****
```

SUMMARY OF STATIONS HAVING POLARITIES IN DISCREPANCY WITH BEST FIT SOLUTION (* DENOTES REVERSED STATION)

STATION	DISCREPANCIES	AGREEMENTS	TOTAL	WEIGHTED ERROR RATE	TOTAL ERROR CONTRIBUTION
CBW	1	0	1	1.000	0.0001
CDU	0	3	3	0.000	0.0000
GAR	1	3	4	0.131	0.0001
GAS	0	1	1	0.000	0.0000
GAX	5	197	202	0.015	0.0008
GBG	5	195	200	0.010	0.0005
GBM	0	1	1	0.000	0.0000
GCM	4	197	201	0.005	0.0002
GCR	5	200	205	0.005	0.0003
GCS	0	14	14	0.000	0.0000
GCV	6	167	173	0.016	0.0006
GCW	0	13	13	0.000	0.0000
GDC	3	53	56	0.025	0.0003
GDX	7	198	205	0.013	0.0006
GGL	2	207	209	0.008	0.0004
GGP	1	36	37	0.003	0.0000
GGPV	9	183	192	0.015	0.0007
GGU	0	21	21	0.000	0.0000
GHC	2	34	36	0.022	0.0002
GHG	6	37	43	0.120	0.0008
GHL	6	78	84	0.055	0.0009
GHV	0	11	11	0.000	0.0000
GMC	2	75	77	0.015	0.0003
GMK	7	156	163	0.024	0.0008
GMM	7	185	192	0.026	0.0012
GMO	5	171	176	0.012	0.0005
GNA	0	1	1	0.000	0.0000
GPM	6	201	207	0.017	0.0010
GRT	5	161	166	0.014	0.0005
GSG	7	196	203	0.012	0.0006
GSM	2	208	210	0.003	0.0001
GSN	2	35	37	0.017	0.0001
GSS	15	154	169	0.053	0.0018
GWK	0	5	5	0.000	0.0000
GWR	1	4	5	0.109	0.0001
JPR	0	1	1	0.000	0.0000
NBP	3	6	9	0.222	0.0004
NBR	3	6	9	0.338	0.0004
NCF	1	2	3	0.586	0.0002
NDH	0	2	2	0.000	0.0000
NFR	1	18	19	0.033	0.0001
NGV	1	4	5	0.202	0.0002
NHB	10	93	103	0.031	0.0007
NHM	0	1	1	0.000	0.0000
NLH	0	1	1	0.000	0.0000
NLN	1	5	6	0.039	0.0000
NMH	5	144	149	0.020	0.0007
NMT	9	124	133	0.042	0.0011
NMW	10	110	120	0.056	0.0013
NOL	1	1	2	0.359	0.0001
NSH	11	150	161	0.037	0.0012
NSP	1	5	6	0.084	0.0001
NTB	2	3	5	0.382	0.0002
NTM	3	7	10	0.316	0.0006
NVA	1	2	3	0.349	0.0002
NVE	1	17	18	0.005	0.0000
NWR	3	8	11	0.217	0.0005
TOTAL	189	4111	4300		

SUMMARY OF HAND-PICKED DATA WITH RESPECT TO BEST FIT SOLUTIONS

QUAL	DISCREPANCIES	AGREEMENTS	TOTAL	WEIGHTED ERROR RATE
0	64	2526	2590	0.0106 **
1	15	595	610	0.0144 **
2	70	682	752	0.0693 **
3	21	170	191	0.0703 **
TOTAL	170	3973	4143	

SUMMARY OF MACHINE-PICKED DATA WITH RESPECT TO BEST FIT SOLUTIONS

QUAL	DISCREPANCIES	AGREEMENTS	TOTAL	WEIGHTED ERROR RATE
0	19	138	157	0.0860 **
1	0	0	0	0.0000
2	0	0	0	0.0000
3	0	0	0	0.0000
TOTAL	19	138	157	

DISTRIBUTION OF SOLUTION MISFIT SCORES

MISFIT SCORE	#
0.000 - 0.025	127
0.025 - 0.050	41
0.050 - 0.075	27
0.075 - 0.100	13
0.100 - 0.125	2
0.125 - 0.150	0
0.150 - 0.175	0
0.175 - 0.200	0
0.200 - 0.225	0
0.225 - 0.250	0
0.250 - 0.275	0
0.275 - 0.300	0
0.300 - 0.325	0
0.325 - 0.350	0
0.350 - 0.375	0
0.375 - 0.400	0
0.400 - 0.425	0
0.425 - 0.450	0
0.450 - 0.475	0
0.475 - 0.500	0

DISTRIBUTION OF SOLUTION DIP RANGES

RANGE	#
0.0	0
5.0	9
10.0	22
15.0	35
20.0	26
25.0	29
30.0	24
35.0	19
40.0	19
45.0	17
50.0	7
55.0	2
60.0	0
65.0	1
70.0	0
75.0	0
80.0	0
85.0	0
90.0	0

DISTRIBUTION OF SOLUTION STRIKE RANGES

RANGE	#
0.0	0
5.0	18
10.0	65
15.0	65
20.0	29
25.0	16
30.0	5
35.0	4
40.0	3
45.0	3
50.0	0
55.0	1
60.0	0
65.0	0
70.0	0
75.0	0
80.0	1
85.0	0
90.0	0

DISTRIBUTION OF SOLUTION RAKE RANGES

RANGE	#
0.0	0
10.0	32
20.0	68
30.0	79
40.0	23
50.0	6
60.0	2

210 EVENTS READ, 210 PROCESSED

B.5 Portion of the FIT FILE for one selected earthquake.

7ØØ711 19 1 3.79 37- 8.29 122- Ø.34 13.4Ø 2.94 29 9Ø 3.Ø Ø.1Ø Ø.3 Ø.7 A 24Ø 5Ø-16Ø Ø.Ø3 23 3.9Ø Ø.59 Ø.ØØ 15 35 2Ø 818

SLIP ANGLE = -19Ø

STRIKE: DIP	115	12Ø	125	13Ø	135	14Ø	145	15Ø	155	16Ø	165	17Ø	175	18Ø	185	19Ø	195	2ØØ	2Ø5
5	565	558	554	543	524	51Ø	495	476	461	441	412	396	38Ø	362	348	338	33Ø	317	3Ø9
1Ø	542	527	515	498	474	458	446	426	4Ø9	382	359	341	327	31Ø	3Ø9	31Ø	312	319	33Ø
15	518	488	456	425	396	369	336	31Ø	286	254	247	258	274	284	299	317	329	346	369
2Ø	5Ø2	456	4Ø5	356	318	288	258	24Ø	199	172	187	215	233	28Ø	312	346	37Ø	395	427
25	5Ø7	433	358	314	283	236	196	164	173	173	191	22Ø	249	277	3Ø5	352	39Ø	433	473
3Ø	5Ø1	436	349	282	24Ø	199	158	146	151	174	192	22Ø	254	288	316	365	398	449	496
35	5Ø8	436	337	269	211	168	128	134	15Ø	173	194	224	251	295	324	366	413	464	519
4Ø	5Ø7	422	348	261	2ØØ	142	95*	13Ø	147	165	199	233	258	3Ø7	34Ø	382	427	482	552
45	5Ø6	418	343	256	192	117	92*	131	148	166	2ØØ	236	263	311	346	391	432	5Ø7	563
5Ø	5Ø7	421	33Ø	257	185	1Ø5	1Ø2	13Ø	148	167	199	237	265	312	35Ø	396	446	5Ø6	565
55	496	421	33Ø	257	182	123	1Ø4	129	147	167	194	237	265	3Ø6	351	4Ø3	45Ø	499	565
6Ø	494	431	334	256	184	13Ø	118	127	146	165	187	235	265	3Ø4	352	41Ø	451	495	563
65	5Ø2	432	342	267	179	134	137	124	144	162	186	231	263	3Ø2	353	41Ø	451	492	56Ø
7Ø	5Ø3	433	345	265	184	145	144	128	142	157	184	226	261	299	36Ø	4Ø5	45Ø	49Ø	555
75	5Ø2	44Ø	347	256	192	151	149	131	139	154	181	217	259	292	36Ø	4Ø1	448	487	546
8Ø	499	435	354	246	192	176	151	139	147	151	178	2Ø8	255	296	357	399	445	483	537
85	494	424	351	241	187	185	151	143	146	147	173	197	251	296	353	395	442	48Ø	525
9Ø	492	421	357	236	181	181	159	142	142	15Ø	167	192	245	294	345	392	438	476	518
95	49Ø	42Ø	357	247	187	186	163	139	13Ø	142	155	184	245	29Ø	337	387	432	472	512

SLIP ANGLE = -18Ø

STRIKE: DIP	115	12Ø	125	13Ø	135	14Ø	145	15Ø	155	16Ø	165	17Ø	175	18Ø	185	19Ø	195	2ØØ	2Ø5
5	567	564	557	55Ø	547	537	518	5Ø5	491	472	458	441	412	395	38Ø	363	349	34Ø	332
1Ø	554	54Ø	525	51Ø	5ØØ	485	468	453	436	416	4ØØ	376	347	328	318	3Ø7	3Ø5	313	321
15	549	515	484	454	426	397	357	329	3Ø3	283	267	264	27Ø	282	289	293	3Ø2	314	328
2Ø	538	483	433	379	327	293	263	236	193	183	19Ø	2ØØ	2Ø5	25Ø	277	3Ø3	325	345	365
25	541	456	384	333	282	228	19Ø	146	152	169	181	194	212	238	261	3ØØ	337	372	4ØØ
3Ø	532	457	374	299	231	183	146	121	123	162	178	188	213	245	267	3Ø8	339	384	418
35	537	453	361	284	196	145	11Ø	1Ø3	117	152	173	187	2Ø7	248	273	3Ø7	352	395	437
4Ø	534	436	368	273	181	112	69*	93*	1Ø6	132	165	184	2Ø9	255	284	321	364	413	47Ø
45	53Ø	429	361	263	167	8Ø*	65*	94*	1Ø6	132	163	184	211	258	29Ø	33Ø	37Ø	441	485
5Ø	528	428	344	263	158	65*	75*	95*	1Ø6	132	16Ø	183	213	259	294	337	388	449	494
55	512	424	342	262	156	84*	78*	95*	1Ø6	132	154	182	214	254	297	347	397	448	5ØØ
6Ø	5Ø7	431	343	261	159	92*	92*	95*	1Ø6	132	148	181	215	254	3ØØ	358	4Ø2	45Ø	5Ø5
65	513	431	344	271	157	97*	112	95*	1Ø6	132	148	179	215	256	3Ø4	364	4Ø7	454	5Ø8
7Ø	511	431	344	269	17Ø	111	121	1Ø8	1Ø5	132	148	178	216	258	315	368	411	457	51Ø
75	5Ø8	436	344	26Ø	178	118	127	112	1Ø5	131	147	175	216	259	319	371	414	46Ø	5Ø8
8Ø	5Ø4	43Ø	35Ø	25Ø	18Ø	146	132	123	115	131	147	172	216	268	321	374	417	462	5Ø7
85	494	42Ø	347	247	177	159	135	13Ø	119	13Ø	146	164	215	274	322	376	419	465	5Ø8
9Ø	489	417	353	243	174	161	146	132	119	137	145	162	215	277	319	378	421	467	51Ø
95	485	415	352	247	181	171	154	132	115	134	14Ø	159	221	278	318	379	423	469	511

SLIP ANGLE = -170

STRIKE: DIP	115	120	125	130	135	140	145	150	155	160	165	170	175	180	185	190	195	200	205
5	575	566	558	556	549	542	540	531	512	500	488	469	456	441	413	395	382	364	352
10	570	553	536	523	508	496	491	478	456	441	424	400	385	362	336	322	313	308	314
15	577	546	515	479	452	422	382	352	316	294	277	276	287	290	284	291	292	291	301
20	567	518	465	400	350	311	268	235	193	174	181	200	209	237	249	274	291	304	324
25	565	488	415	352	304	241	182	132	141	149	159	183	205	216	225	261	291	321	349
30	551	488	402	315	251	193	131	99	106	136	149	171	197	215	228	263	285	325	362
35	553	479	387	297	216	152	90*	75*	96*	121	138	164	184	210	232	258	294	331	376
45	539	450	379	269	180	78*	35*	59*	81*	94*	121	153	172	202	241	273	305	373	423
50	532	446	357	266	169	58*	45*	58*	81*	93*	122	152	172	200	244	279	326	382	435
55	513	438	350	263	165	69*	48*	58*	81*	93*	119	151	172	193	248	289	339	384	444
60	506	443	343	261	166	77*	61*	57*	82*	93*	115	151	172	193	252	302	349	390	449
65	509	441	340	269	163	82*	81*	65*	83*	94*	117	151	173	195	258	311	357	398	456
70	507	438	339	265	173	97*	90*	81*	84*	95*	120	151	175	198	274	318	365	407	464
75	503	442	338	256	180	105	97*	89*	86*	96*	122	151	176	202	282	325	373	419	469
80	498	435	343	246	180	134	103	102	98*	97*	125	150	178	221	289	333	381	429	475
85	491	426	342	244	175	148	108	113	104	105	128	145	180	230	295	342	388	440	485
90	486	417	349	242	169	155	123	120	108	119	131	148	183	250	297	356	398	450	495
95	482	412	347	245	176	166	140	126	110	126	132	149	194	260	303	366	409	461	506

SLIP ANGLE = -160

STRIKE: DIP	115	120	125	130	135	140	145	150	155	160	165	170	175	180	185	190	195	200	205
5	584	576	566	557	550	548	542	536	535	526	508	497	485	467	456	441	414	397	384
10	594	571	551	534	518	506	497	485	477	464	442	423	407	382	370	357	330	323	323
15	614	575	542	510	481	444	404	371	336	309	282	282	293	296	299	300	289	290	295
20	605	546	491	430	375	327	287	247	192	172	180	190	197	229	248	262	265	278	292
25	594	510	435	380	328	255	198	139	129	132	146	162	182	202	218	240	260	282	301
30	571	503	419	342	273	203	145	103	88*	113	131	144	167	193	210	232	248	277	302
35	557	486	397	320	233	159	101	74*	73*	91*	114	129	146	183	202	218	251	273	298
40	545	454	388	300	209	117	45*	49*	51*	58*	95*	113	128	169	190	200	238	276	324
45	532	442	374	273	171	75*	34*	37*	46*	52*	88*	107	127	157	182	204	238	300	339
50	521	434	351	269	154	52*	36*	30A	37*	43*	77*	98	125	156	184	200	258	310	357
55	500	425	346	265	151	71*	39*	32*	39*	50*	74*	99	129	151	187	217	270	314	371
60	492	430	344	260	153	79*	51*	34*	40*	56*	71*	100	133	155	191	237	283	324	385
65	494	428	342	263	151	84*	66*	36*	42*	62*	75*	107	138	159	196	252	296	336	399
70	493	427	340	252	163	98*	71*	50*	44*	66*	78*	115	143	164	213	264	309	356	415
75	490	431	339	243	172	104	80*	58*	54*	71*	82*	120	148	170	224	276	322	373	433
80	488	426	342	235	175	128	88*	71*	73*	76*	87*	124	153	185	242	289	337	390	449
85	490	428	338	237	174	136	96*	82*	84*	81*	103	124	160	198	254	308	359	410	468
90	488	427	351	245	183	153	115	101	94*	95*	114	132	167	211	263	324	379	437	488
95	485	422	351	250	186	168	135	123	109	115	123	140	183	234	277	342	399	457	510

SLIP ANGLE = -150

STRIKE: DIP	115	120	125	130	135	140	145	150	155	160	165	170	175	180	185	190	195	200	205
5	600	585	575	567	558	549	542	541	535	530	530	522	505	495	484	467	456	443	417
10	627	598	574	550	531	515	503	493	481	469	462	444	423	403	391	374	363	358	341
15	659	618	578	537	506	471	427	388	346	315	293	292	296	301	304	303	304	306	304
20	635	577	518	450	393	351	307	256	203	176	181	190	192	215	232	251	263	270	278
25	616	531	442	382	331	273	212	143	137	132	132	144	165	179	194	221	244	259	271
30	590	522	423	343	274	212	146	92*	81*	103	105	114	140	158	170	196	214	239	260
35	581	503	400	322	239	173	105	69*	70*	77*	81*	91*	111	139	157	184	213	234	265
40	570	476	396	306	222	137	50*	51*	52*	46*	63*	76*	95*	125	146	175	201	235	286
45	554	454	382	285	199	102	40*	54*	53*	41*	61*	74*	96*	124	153	178	200	249	290
50	539	440	357	280	187	83*	52*	55*	52*	42*	58*	73*	98	122	157	182	217	256	300
55	512	426	350	275	181	93*	56*	56*	50*	43*	54*	79*	100	117	162	191	227	262	311
60	493	428	345	271	179	88*	68*	55*	40*	45*	49*	83*	103	126	167	207	237	276	323
65	490	424	343	275	171	92*	84*	53*	42*	46*	51*	87*	106	135	174	217	248	289	336
70	486	422	340	269	170	105	91*	58*	43*	47*	61*	90*	111	144	192	228	267	303	351
75	482	425	339	256	172	112	95*	56*	45*	49*	69*	93*	126	152	203	239	284	319	376
80	480	420	344	236	173	136	94*	68*	56*	60*	76*	95*	137	169	214	261	300	338	406
85	485	424	346	233	172	147	92*	79*	63*	70*	83*	108	146	185	226	280	318	374	449
90	483	424	349	244	185	156	110	88*	81*	86*	93*	121	157	200	246	298	341	413	492
95	480	421	348	250	193	167	134	108	104	105	114	133	177	216	266	319	385	461	527

SLIP ANGLE = -140

STRIKE: DIP	115	120	125	130	135	140	145	150	155	160	165	170	175	180	185	190	195	200	205
5	626	611	594	576	565	558	549	541	535	535	530	525	526	520	504	494	484	468	458
10	656	638	603	575	551	529	514	498	484	475	463	447	440	423	405	394	383	374	373
15	708	666	604	564	535	493	446	407	359	321	292	289	303	309	305	308	308	309	314
20	706	632	547	473	401	351	305	263	209	175	185	190	191	210	221	234	248	260	270
25	682	577	484	418	351	281	218	154	138	122	125	130	140	152	171	190	211	233	249
30	645	560	464	382	297	234	172	126	106	109	112	113	122	137	152	175	192	222	243
35	613	529	439	358	262	198	135	104	94*	92*	97*	98*	101	122	141	160	186	213	240
40	597	497	428	336	244	163	84*	83*	72*	64*	78*	80*	82*	105	127	146	169	211	253
45	575	478	411	301	217	129	80*	83*	66*	64*	75*	72*	81*	102	127	147	170	221	251
50	547	462	379	290	203	111	89*	82*	67*	64*	70*	71*	81*	99	129	149	189	228	258
55	515	444	361	284	196	120	90*	76*	67*	63*	63*	70*	81*	92*	131	156	201	229	267
60	501	440	347	278	193	122	95*	69*	66*	61*	49*	69*	81*	101	134	177	211	238	276
65	498	424	342	279	186	121	101	68*	64*	57*	50*	69*	82*	107	139	192	222	249	292
70	490	417	338	272	189	126	100	75*	61*	47*	51*	68*	93*	112	165	205	234	262	312
75	475	417	335	259	189	117	103	76*	52*	48*	52*	68*	100	119	182	218	248	287	328
80	471	411	339	246	179	137	105	80*	55*	50*	54*	81*	107	149	197	233	266	311	358
85	477	417	343	239	169	148	105	76*	61*	52*	69*	84*	127	171	211	251	298	340	403
90	475	418	355	241	184	163	105	84*	67*	72*	81*	103	144	190	224	284	328	387	471
95	472	415	342	247	193	165	132	107	92*	99	98*	124	170	210	258	314	374	456	540

SLIP ANGLE = -130

STRIKE:	115	120	125	130	135	140	145	150	155	160	165	170	175	180	185	190	195	200	205
DIP																			
5	651	641	616	600	583	566	556	550	542	534	529	529	526	522	524	519	504	495	486
10	697	665	629	600	574	548	525	500	492	477	463	450	439	423	421	414	396	394	392
15	762	720	675	621	557	510	454	408	365	325	287	284	296	301	309	315	311	315	321
20	755	696	631	527	442	387	330	273	211	177	181	176	177	187	202	217	228	240	254
25	714	642	563	467	395	320	242	176	159	147	142	138	148	155	168	187	204	218	234
30	677	619	521	429	343	273	196	149	131	135	131	123	133	142	146	164	182	204	223
35	647	576	485	396	308	236	159	127	120	119	116	109	114	127	133	146	169	192	216
40	624	525	459	370	287	198	108	106	99	92*	96*	92*	95*	100	115	128	147	183	224
45	[illegible]	[illegible]	[illegible]	[illegible]	254	153	[illegible]	[illegible]	99	91*	89*	90*	93*	103	113	127	143	[illegible]	217
50	543	475	395	320	233	133	113	107	99	89*	83*	89*	92*	94*	112	127	159	194	226
55	500	449	373	309	213	138	114	106	97*	84*	78*	87*	89*	88*	111	130	168	194	235
60	491	435	362	296	204	138	119	103	92*	75*	73*	84*	84*	88*	112	143	177	202	245
65	486	422	352	280	194	137	128	99	83*	74*	71*	79*	79*	89*	113	158	186	219	256
70	470	418	333	270	195	144	128	97*	75*	71*	67*	67*	79*	90*	128	170	203	234	270
75	464	408	327	256	194	144	123	87*	72*	67*	55*	66*	80*	93*	148	183	223	252	286
80	458	396	329	242	188	152	111	91*	74*	56*	54*	65*	82*	117	162	211	243	283	338
85	463	404	333	238	176	145	111	93*	62*	52*	55*	60*	99	138	192	234	280	334	392
90	462	406	347	245	178	161	117	80*	64*	60*	58*	80*	119	176	213	275	333	392	465
95	460	404	338	239	188	174	128	105	84*	78*	82*	108	161	205	257	331	393	474	556

B.6 Graphic output from program FPPLOT for selected earthquakes.

Note the multiple solutions for event 840909.

Note the low value of STDR for event 750426.

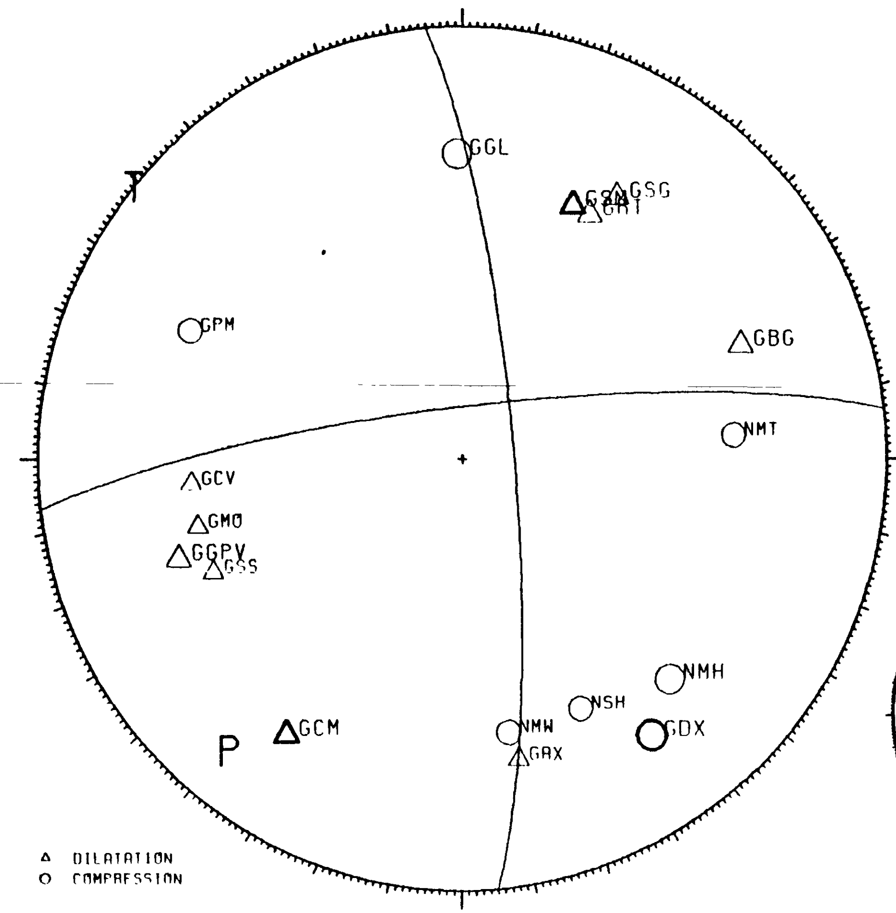
GEYSERS, CALIFORNIA EARTHQUAKES

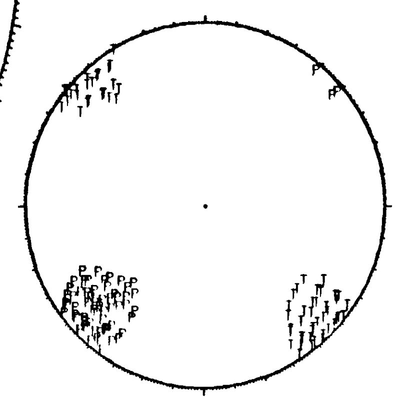
840909 2 8 37.03 38-47.05 122-46.38 1.32 2.54 17 51 2.0 0.03 0.1 0.1 A

85 80-170 0.03 17 4.68 0.58 0.00 20 20 30 B-B

DISCREPANT OBSERVATIONS

STAT	DIST	AZM	AIN	PRMK
GAX	8.3	169	60	IPU1
NMW	26.3	170	54	IPU1



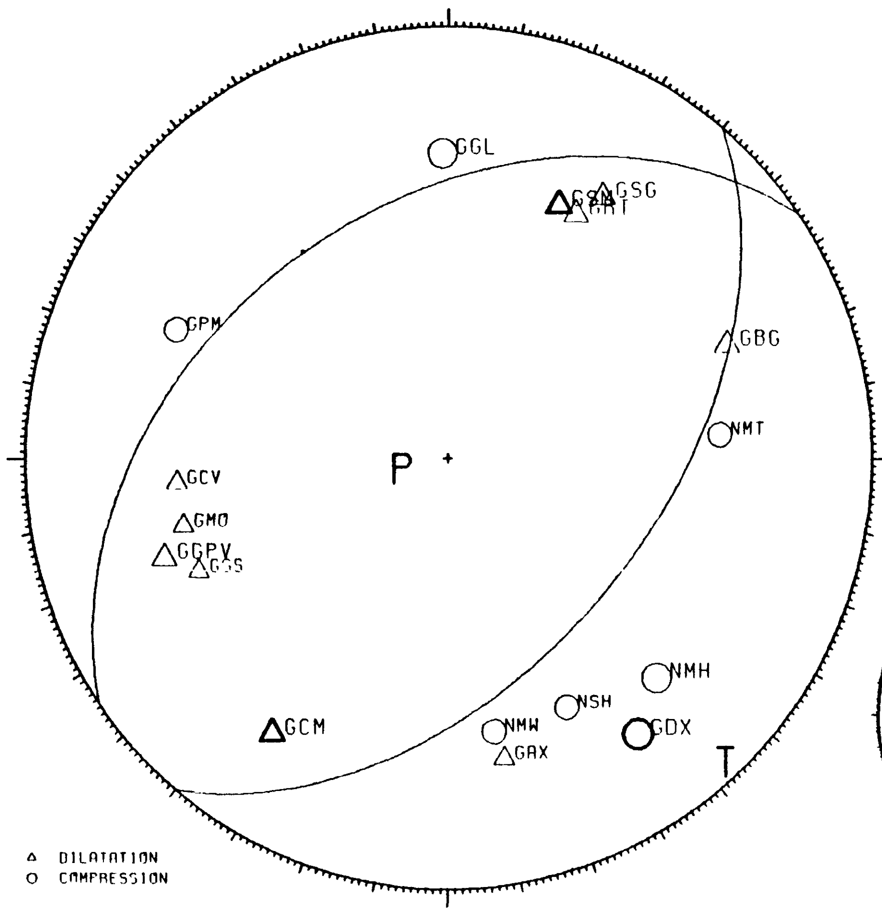


△ DILATATION
○ COMPRESSION

GEYSERS, CALIFORNIA EARTHQUAKES

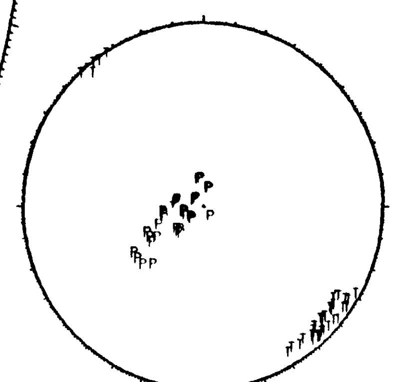
840909 2 8 37.03 38-47.05 122-46.38 1.32 2.54 17 51 2.0 0.03 0.1 0.1 A

130 50-100 0.06 17 4.68 0.50 0.00 15 10 10 B←A*



DISCREPANT OBSERVATIONS

STAT	DIST	AZM	AIN	PRMK
GAX	8.3	169	60	IPD1

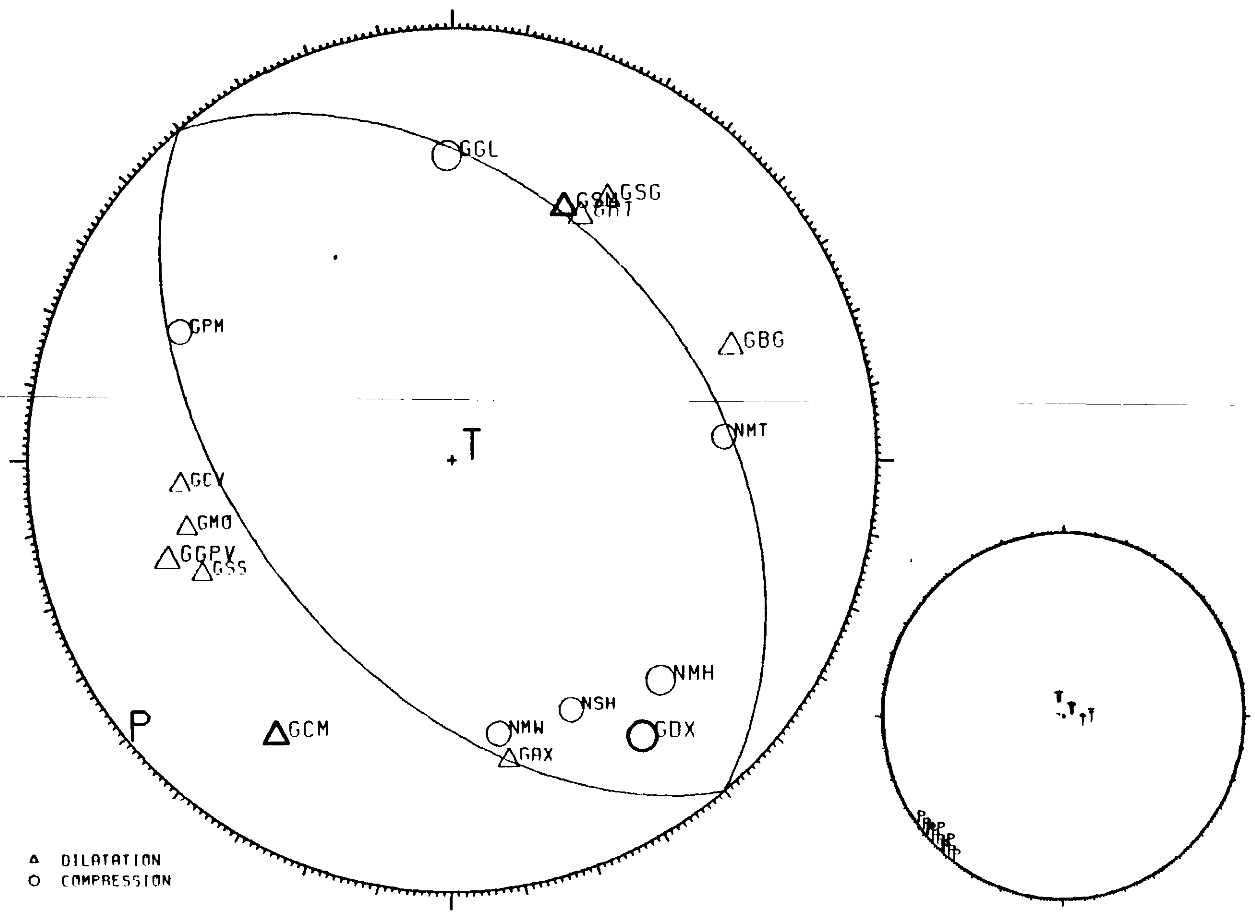


△ DILATATION
○ COMPRESSION

GEYSERS, CALIFORNIA EARTHQUAKES

840909 2 8 37.03 38-47.05 122-46.38   1.32   2.54 17 51 2.0 0.03 0.1 0.1 A

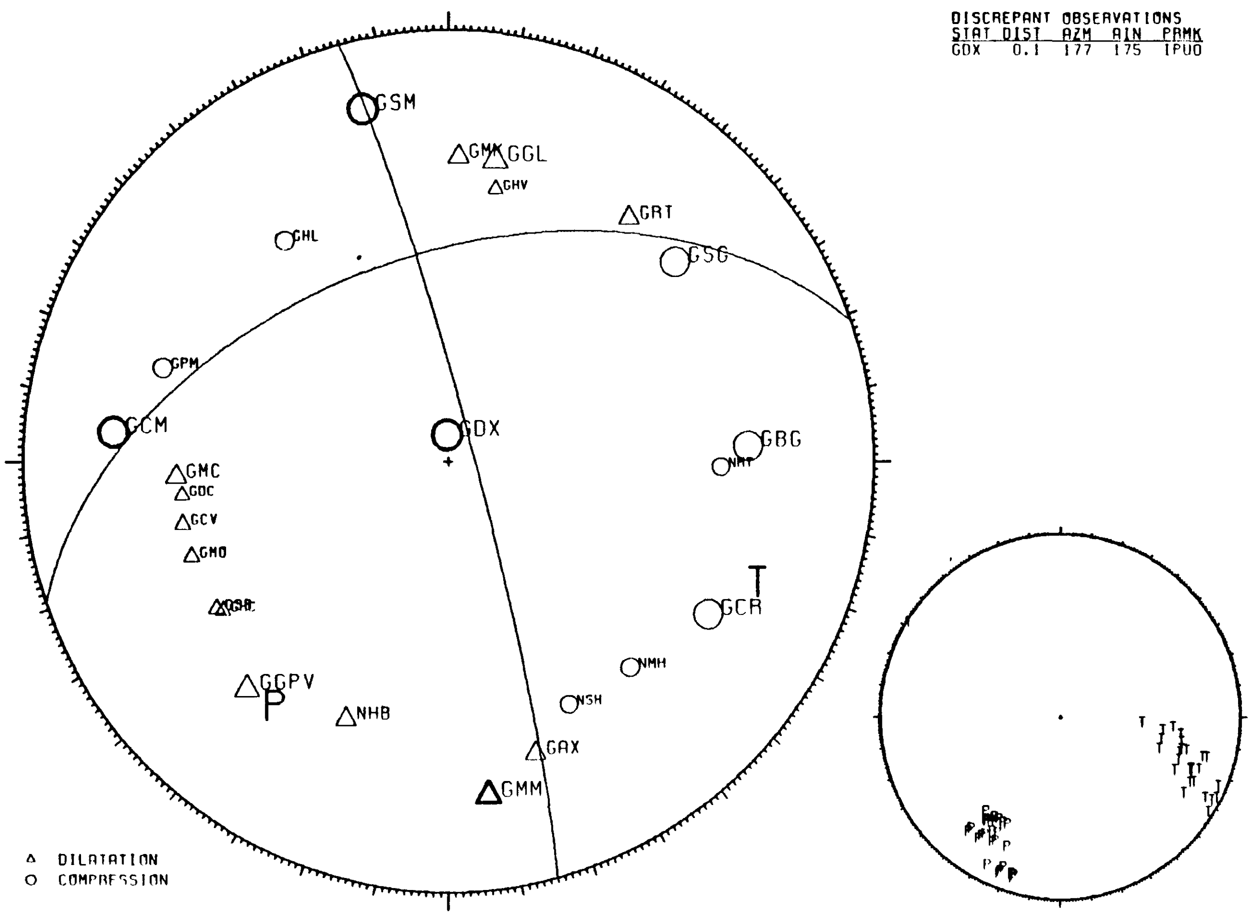
230 50 90 0.00 17 4.68 0.42 0.00 10 0 10 A+A*



GEYSERS, CALIFORNIA EARTHQUAKES

840410 4 2 2.49 38-48.52 122-47.63 1.38 2.95 25 38 0.0 0.05 0.1 0.1 A

75 85-140 0.02 25 4.09 0.56 0.00 10 15 20 A-A



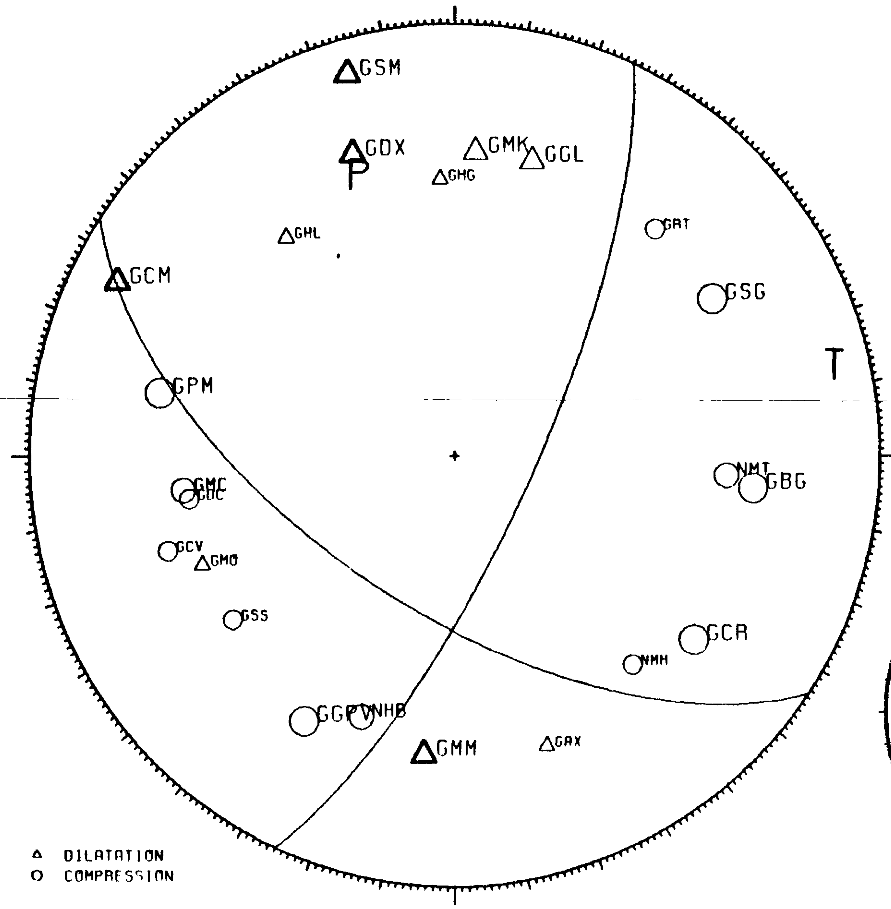
GEYSERS, CALIFORNIA EARTHQUAKES

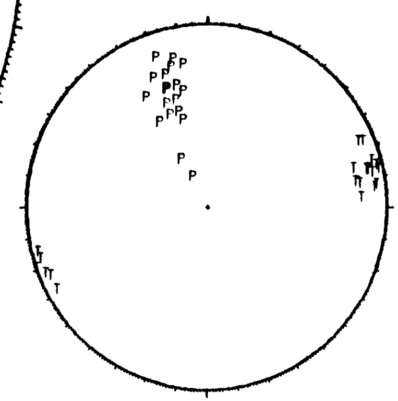
850509 314 18.95 38-49.43 122-48.05 0.96 1.75 24 40 2.0 0.07 0.2 0.2 A

115 75 -30 0.04 23 4.29 0.62 0.00 10 20 20 B←A

DISCREPANT OBSERVATIONS

STAT	DIST	AZM	AIN	PRMK
GMO	32.3	247	54	EPD2





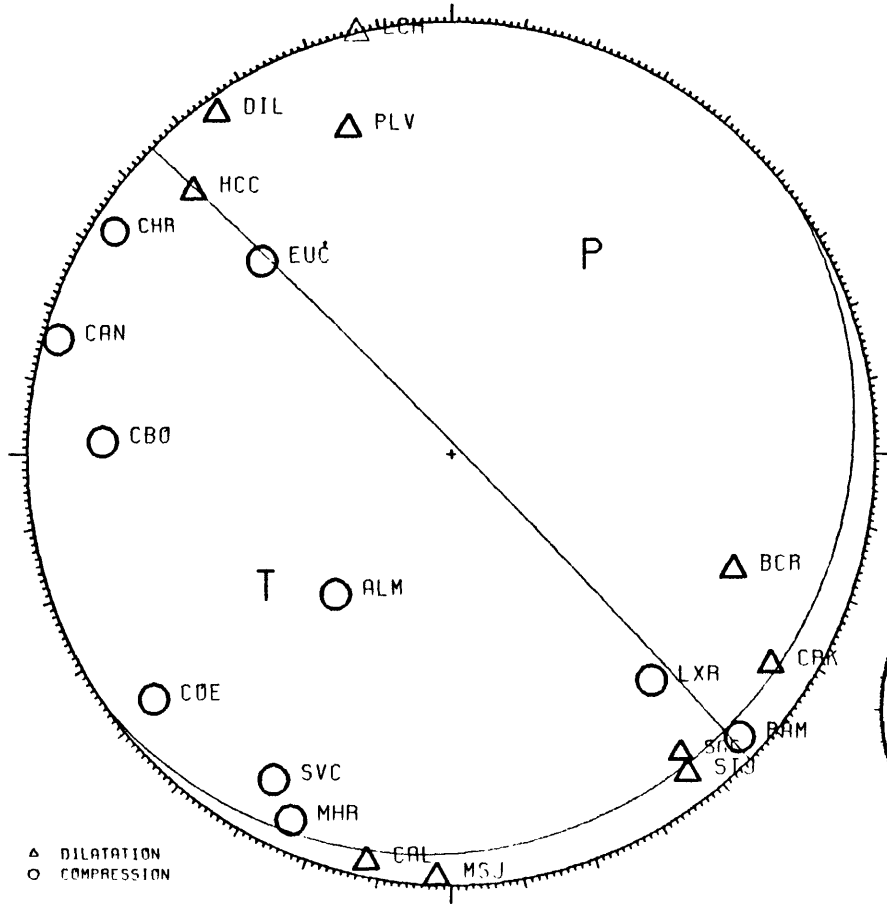
△ DILATATION
○ COMPRESSION

750426 9 6 57.89 37- 7.01 121-53.43 10.45 1.57 33 83 6.0 0.08 0.2 0.4 A

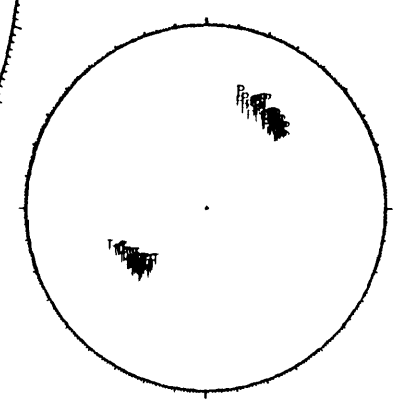
145 10 10 0.03 20 3.88 0.37 0.00 10 10 10 B←A

DISCREPANT OBSERVATIONS

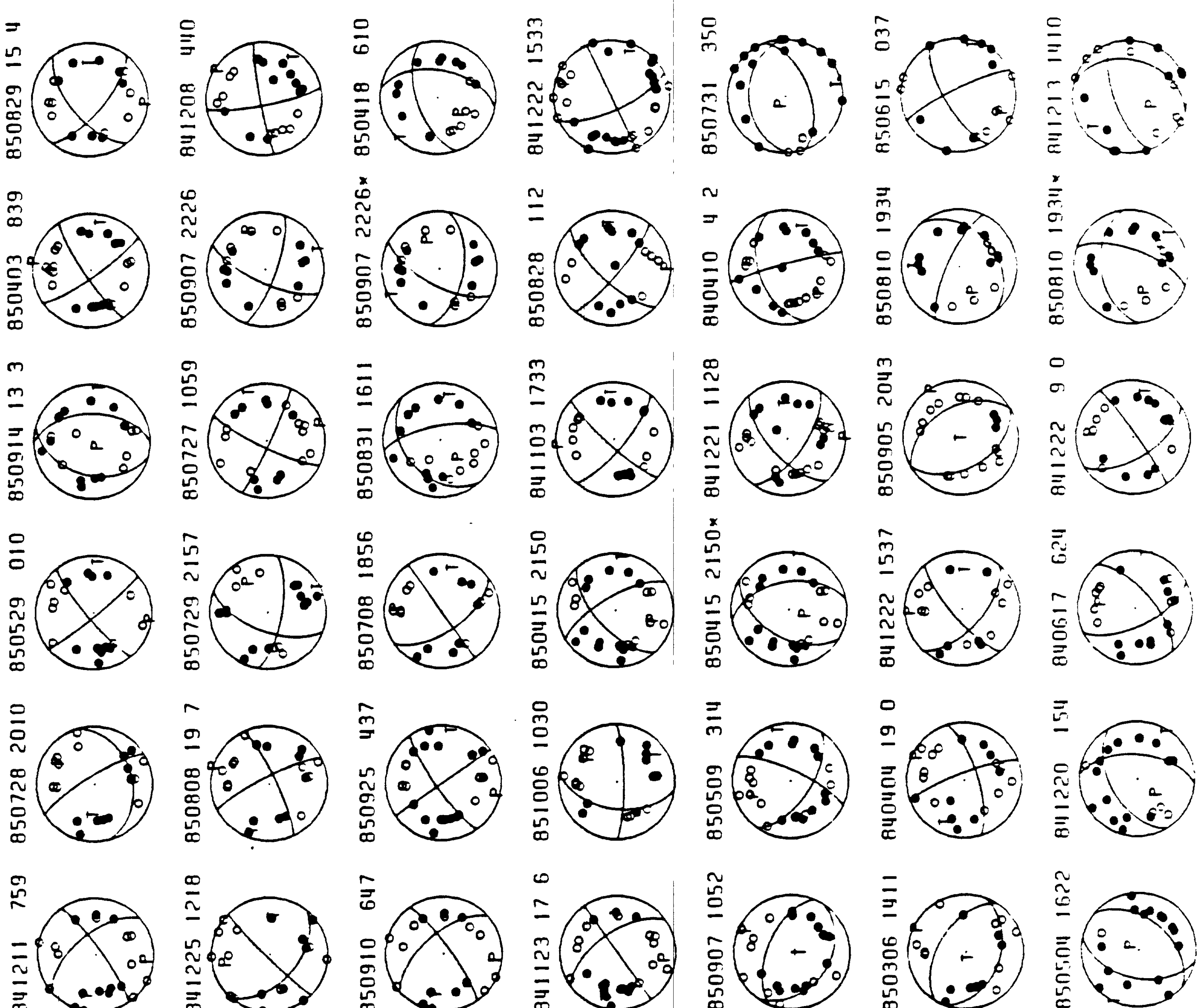
STAT	DIST	AZM	AIN	PRMK
HCC	21.2	135	105	IPD0
SGC	23.2	322	103	IP-1



△ DILATATION
○ COMPRESSION



B.7 Graphic output from program FPPAGE for the first 42 solutions listed in section B.2.



APPENDIX C

C.1 FPFIT FORTRAN listing

C.2 FPPLOT FORTRAN listing

C.3 FPPAGE FORTRAN listing

```
      PROGRAM FPFIT
C
C
C     VERSION 1.0  -  OCTOBER 31, 1985
C
C     PURPOSE:       CALCULATE DOUBLE-COUPLE FAULT PLANE SOLUTIONS FROM P-WAVE FIRST MOTIONS (SEE REASENBERG, P.  AND
C                    D. OPPENHEIMER, FPFIT, FPPLOT AND FPPAGE: FORTRAN COMPUTER PROGRAMS FOR CALCULATING AND DISPLAYING
C                    EARTHQUAKE FAULT-PLANE SOLUTIONS, U.S. GEOLOGICAL SURVEY OPEN-FILE REPORT 85-???).
C
C
C     INPUT FILE:    1.  A HYPO71 LISTING FILE WHICH IS READ ON LOGICAL UNIT IUNIT (=2), (SEE LEE AND LAHR, 1975, HYPO71
C                        (REVISED): A COMPUTER PROGRAM FOR DETERMINING HYPOCENTER, MAGNITUDE AND FIRST
C                        MOTION PATTERN OF LOCAL EARTHQUAKES, U. S. GEOLOGICAL SURVEY OPEN-FILE REP. 75-311.).
C
C                    2.  A PARAMETER CONTROL FILE WHICH IS READ ON LOGICAL UNIT CUNIT (=1)
C
C     REQUIRED ROUTINES: ALL ROUTINES ARE ENCLOSED
C
C     DEPARTURES FROM FORTRAN-77 STANDARD:
C                    1. KEYWORDS "READONLY" AND "CARRIAGECONTROL = LIST" IN OPEN STATEMENTS
C                    2. EMBEDDED COMMENTS PREFACED WITH AN EXCLAMATION MARK (!) FOLLOWING VARIABLE DECLARATIONS
C
C      OUTPUT:       1. AN ASCII FILE OF HYPO71 SUMMARY CARDS EXTENDED WITH FAULT PLANE SOLUTION PARAMETERS ON LOGICAL UNIT SUNIT
C                       (=4).  THIS FILE SERVES AS INPUT TO PROGRAMS "QPLOT" AND "PLOTPT".
C                    2. AN ASCII FILE CONSISTING, FOR EACH EARTHQUAKE, OF THE HYPO71 EXTENDED SUMMARY CARD, FOLLOWED BY
C                       NEIGHBORING SOLUTIONS (WITHIN 90% CONFIDENCE LIMITS), FOLLOWED BY INDIVIDUAL P-PHASE INFORMATION,
C                       ON LOGICAL UNIT PUNIT (=3). THIS FILE SERVES AS INPUT TO PROGRAMS "PLOTFM" AND "FPPAGE".
C                    3. AN OPTIONAL ASCII FILE OF THE MINIMIZED FIT FUNCTION ABOUT THE BEST SOLUTION ON LOGICAL UNIT FUNIT (=9)
C                    4. AN ASCII FILE DESCRIBING ANY ERRORS IN THE CONTROL FILE, HYPO71 FILE, PRESENCE OF MULTIPLE MECHANISMS,
C                       A SUMMARY OF POLARITY DISCREPANCIES BY STATION AND READING QUALITY, AND THE DISTRIBUTION OF STRIKE, DIP, AND
C                       RAKE UNCERTAINTIES ON LOGICAL UNIT EUNIT (=8)
C
C      AUTHORS:      PAUL REASENBERG AND DAVID OPPENHEIMER, U.S.G.S. IN MENLO PARK.  SOME OF THE ROUTINES
C                    WERE ADAPTED FROM CODE WRITTEN BY JOHN LAHR, BRUCE JULIAN, AND FRED KLEIN.
C                    MARK MATTHEWS, STANFORD UNIVERSITY, PROVIDED ASSISTANCE IN THE ERROR PROPAGATION ANALYSIS.
C
C
C
      INTEGER          CUNIT                          ! LOGICAL UNIT # OF INPUT CONTROL FILE
      INTEGER          EUNIT                          ! LOGICAL UNIT # OF OUTPUT OF ERROR MESSAGES
      INTEGER          FUNIT                          ! LOGICAL UNIT # OF OUTPUT OF FIT LISTING FOR ALL STRIKES, DIPS
      INTEGER          IUNIT                          ! LOGICAL UNIT # OF HYPO71 LISTING FILE (INPUT FILE)
      INTEGER          MXDIP                          ! MAXIMUM # OF DIP INCREMENTS PERMITTED
      INTEGER          MXQUAL                         ! MAXIMUM # OF QUALITIES PERMITTED
      INTEGER          MXRAKE                         ! MAXIMUM # OF RAKE INCREMENTS PERMITTED
      INTEGER          MXSLNS                         ! MAXIMUM # OF MULTIPLE SOLUTIONS PERMITTED
      INTEGER          MXSTAT                         ! MAXIMUM # OF STATIONS PERMITTED
      INTEGER          MXSTRK                         ! MAXIMUM # OF STRIKE INCREMENTS PERMITTED
      INTEGER          PUNIT                          ! LOGICAL UNIT # OF OUTPUT OF EXTENDED SUMMARY AND RAY PARAMETERS
                                                      ! FOR USE IN PLOTTING FOCAL MECH. DIAGRAMS WITH PLOTFM
      INTEGER          SUNIT                          ! LOGICAL UNIT # OF OUTPUT OF EXTENDED SUMMARY CARDS
C
      PARAMETER (CUNIT = 1, EUNIT = 8, FUNIT = 9, IUNIT = 2, MXDIP = 19,
     & MXQUAL = 8, MXRAKE = 18, MXSLNS = 20, MXSTAT = 1000, MXSTRK = 19,
     & PUNIT = 3, SUNIT = 4)
C
      REAL             AERR                           ! ALLOWABLE DIFFERENCE BETWEEN CORRESPONDING ANGLES OF COMPLIMENTARY PLANES
      REAL             AIN(MXSTAT)                    ! RAY INCIDENCE ANGLES IN DEGREES
      REAL             AINR                           ! AIN CONVERTED TO RADIANS
      REAL             AZ(MXSTAT)                     ! RAY AZIMUTH ANGLES (CORRESPONDING TO AIN)
      REAL             AVWT                           ! MEAN OBSERVATIONAL WEIGHT OF DATA USED IN SOLUTION
      REAL             AZR                            ! AZ CONVERTED TO RADIANS
      CHARACTER*1      BDFLAG                         ! SIGNALS POLARITY DISCREPANCY WITH BEST FIT SOLUTION
      LOGICAL          BEST                           ! FLAG: TRUE=BEST SOLUTION FOR EVENT
```

```
REAL           BOT(MXDIP,MXSTRK,MXRAKE)      ! SUM OF PRODUCT OF OBSERVED AND PREDICTED WEIGHTS
REAL           COEF(MXSTAT,6)                ! COEFICIENTS BY WHICH MOMENT TENSOR MULTIPLIED TO GIVE P RADIATION PATTERN
LOGICAL        COMPL                         ! FUNCTION
REAL           DA2                           ! DIP ANGLE OF AUXILLIARY SOLUTION
REAL           DD2                           ! DIP DIRECTION OF AUXILLIARY SOLUTION
REAL           DDELC                         ! FAULT DIP INCREMENT IN DEGREES FOR COARSE SEARCH
REAL           DDELF                         ! FAULT DIP INCREMENT IN DEGREES FOR FINE SEARCH
REAL           DEL(MXDIP)                    ! FAULT DIP ANGLE IN DEGREES
REAL           DELC(MXDIP)                   ! FAULT DIP ANGLE IN DEGREES FOR COARSE SEARCH
REAL           DELØC                         ! INITIAL FAULT DIP ANGLE IN DEGREES FOR COARSE SEARCH
REAL           DELØF                         ! INITIAL FAULT DIP ANGLE IN DEGREES FOR FINE SEARCH
REAL           DELIF                         ! TERMINATING FAULT DIP ANGLE IN DEGREES FOR FINE SEARCH
REAL           DFITC                         ! INCREMENT TO COARSE FIT FUNCTION
REAL           DIP                           ! DIP ANGLE OF BEST SOLUTION
REAL           DIST(MXSTAT)                  ! EPICENTRAL DISTANCE
REAL           DISTMX                        ! MAXIMUM PERMITTED EPICENTRAL DISTANCE
REAL           DLAMC                         ! FAULT RAKE INCREMENT IN DEGREES FOR COARSE SEARCH
REAL           DLAMF                         ! FAULT RAKE INCREMENT IN DEGREES FOR FINE SEARCH
REAL           DPHIC                         ! FAULT STRIKE INCREMENT IN DEGREES FOR COARSE SEARCH
REAL           DPHIF                         ! FAULT STRIKE INCREMENT IN DEGREES FOR FINE SEARCH
REAL           ERATE(MXQUAL)                 ! ASSUMED WEIGHTED ERROR RATES FOR EACH DATA CLASS
CHARACTER*8Ø   EVENT                         ! SUMMARY CARD
CHARACTER*52   EVFIT                         ! DUMMY CHARACTER STRING TO HOLD FIT VALUES ON OUTPUT
LOGICAL        FIRST                         ! FLAG: TRUE=FIRST TIME INTO SUBROUTINE SEARCH
REAL           FIT9Ø                         ! 9Ø % CONFIDENCE LIMIT OF FIT IN FINE SEARCH
REAL           FIT(MXDIP,MXSTRK,MXRAKE)      ! SOLUTION FIT; WEIGHTED MEASURE OF AGREEMENT BETWEEN OBS, PRED POLARITIES
REAL           FITLIM                        ! UPPER BOUND ON "GOOD" SOLUTIONS; 5% ABOVE FITMIN
REAL           FITMIN                        ! FIT OF BEST SOLUTION CORRESPONDING TO FIT(J1, N1, M1)
CHARACTER*1    FLAG(MXDIP,MXSTRK,MXRAKE)     ! OUTPUT STRING: IF FIT < FITLIM THEN '*', OTHERWISE BLANK
REAL           FMAGMN                        ! MINIMUM PERMITTED MAGNITUDE
LOGICAL        FPINP                         ! FUNCTION
CHARACTER*3    FTQUAL                        ! SOLUTION QUALITY: CONCATENATED FIT QUALITY, RANGE QUALITY
REAL           GFIT(MXDIP,MXSTRK,MXRAKE)     ! FITS OF "GOOD" SOLUTIONS FOUND IN COARSE SEARCH
INTEGER        I                             ! LOOP INDEX
INTEGER        ID                            ! LOOP INDEX OVER NDST
INTEGER        IDIP                          ! DIP ANGLE OF NEARBY SOLUTIONS
INTEGER        IDIP1                         ! DIP ANGLE OF BEST FIT
INTEGER        IDIPDR                        ! DIP DIRECTION OF NEARBY SOLUTIONS
INTEGER        IDST(MXSLNS,3)                ! INDICES OF DISTINCT SOLUTIONS FOUND BY HHOG
INTEGER        IDPDR1                        ! DIP DIRECTION OF BEST FIT
INTEGER        IDRNG                         ! DIP RANGE VARIATION FOR SOLUTIONS WITH FIT<FITLIM
INTEGER        IERR                          ! I/O ERROR CODE ON OPEN
INTEGER        IEVP                          ! NUMBER OF EVENTS PROCESSED
INTEGER        IEVR                          ! NUMBER OF EVENTS READ
INTEGER        IFIT(MXSTRK)                  ! INTEGER CONVERSION OF FIT*1ØØ FOR PRINTER OUTPUT
INTEGER        IGOOD(MXDIP*MXSTRK*MXRAKE,4)  ! ARRAY OF INDICES OF "GOOD" SOLUTIONS FOUND IN COARSE SEARCH
INTEGER        IJEFF                         ! FLAG: 1(Ø)=DO (NOT) USE PHASE DATA WEIGHTED OUT BY JEFFREY'S WEIGHTING
INTEGER        IND(MXSTAT)                   ! POINTER ARRAY TO SORTED ORDER
INTEGER        INDEX                         ! BIN INDEX INTO NDRNG,NSRNG,NRRNG,NFIT
INTEGER        IPRNT                         ! FLAG: I(Ø)=DO (NOT) PRINT OUT FIT PARAMETERS
INTEGER        IPWT                          ! WEIGHT ASSIGNED TO P ARRIVAL
INTEGER        IRES                          ! FLAG: Ø(1)=(UN)RESTRICTED SEARCH
INTEGER        IRRNG                         ! RAKE RANGE VARIATION FOR SOLUTIONS WITH FIT<FITLIM
INTEGER        ISLIP                         ! RAKE OF NEARBY SOLUTIONS
INTEGER        ISLIP1                        ! RAKE OF BEST FIT
INTEGER        ISRNG                         ! STRIKE RANGE VARIATION FOR SOLUTIONS WITH FIT<FITLIM
INTEGER        J                             ! LOOP INDEX OVER DIP
INTEGER        J1                            ! DIP INDEX OF BEST SOLUTION
INTEGER        JMAX                          ! LARGEST DIP INDEX OF SOLUTION WITH FIT <=FITLIM ABOUT (J1, N1, M1)
INTEGER        JMIN                          ! SMALLEST DIP INDEX OF SOLUTION WITH FIT <=FITLIM ABOUT (J1, N1, M1)
INTEGER        K                             ! LOOP INDEX
CHARACTER*4    KILSTA(MXSTAT)                ! IGNORED STATION NAMES
INTEGER        L                             ! LOOP INDEX OVER MOMENT TENSOR
CHARACTER*122  LINE                          ! OUTPUT STRING OF NEARBY SOLUTIONS ORIENTATIONS
```

```
      INTEGER          M                    ! LOOP INDEX OVER RAKE
      INTEGER          M1                   ! RAKE INDEX OF BEST SOLUTION
      INTEGER          MINOBS               ! MINIMUM NUMBER OF OBSERVATIONS REQUIRED
      INTEGER          MMAX                 ! LARGEST RAKE INDEX OF SOLUTION WITH FIT <=FITLIM ABOUT (J1, N1, M1)
      INTEGER          MMIN                 ! SMALLEST RAKE INDEX OF SOLUTION WITH FIT <=FITLIM ABOUT (J1, N1, M1)
      INTEGER          N                    ! LOOP INDEX OF STRIKE
      INTEGER          N1                   ! STRIKE INDEX OF BEST SOLUTION
      INTEGER          NDELC                ! NUMBER OF FAULT DIP INCREMENTS FOR COARSE SEARCH
      INTEGER          NDELF                ! NUMBER OF FAULT DIP INCREMENTS FOR FINE SEARCH
      INTEGER          NDELFDF              ! DEFAULT NUMBER OF FAULT DIP INCREMENTS FOR FINE SEARCH
      INTEGER          NDRNG(MXDIP)         ! NUMBER OF DIP SOLUTION RANGES BINNED INTO DDELF DEGREE INCREMENTS
      INTEGER          NDST                 ! NUMBER OF DISTINCT SOLUTIONS FOUND BY HHOG
      INTEGER          NEV                  ! NUMBER OF EVENTS TO PROCESS
      INTEGER          NFIT(2Ø)             ! NUMBER OF SOLUTIONS BINNED INTO .Ø25 FIT INCREMENTS
      INTEGER          NG                   ! NUMBER OF "GOOD" SOLUTIONS FOUND IN COARSE SEARCH
      INTEGER          NKIL                 ! NUMBER OF IGNORED STATIONS
      INTEGER          NLAMC                ! NUMBER OF FAULT RAKE INCREMENTS FOR COARSE SEARCH
      INTEGER          NLAMF                ! NUMBER OF FAULT RAKE INCREMENTS FOR FINE SEARCH
      INTEGER          NLAMFDF              ! DEFAULT NUMBER OF FAULT RAKE INCREMENTS FOR FINE SEARCH
      INTEGER          NMAX                 ! LARGEST STRIKE INDEX OF SOLUTION WITH FIT <=FITLIM ABOUT (J1, N1, M1)
      INTEGER          NMIN                 ! SMALLEST STRIKE INDEX OF SOLUTION WITH FIT <=FITLIM ABOUT (J1, N1, M1)
      INTEGER          NPHIC                ! NUMBER OF FAULT STRIKE INCREMENTS FOR COARSE SEARCH
      INTEGER          NPHIF                ! NUMBER OF FAULT STRIKE INCREMENTS FOR FINE SEARCH
      INTEGER          NPHIFDF              ! DEFAULT NUMBER OF FAULT STRIKE INCREMENTS FOR FINE SEARCH
      INTEGER          NR                   ! -1=EOF, Ø=SKIP EVENT, NR>Ø => NUMBER OF STATIONS
      INTEGER          NREV                 ! NUMBER OF REVERSED STATIONS
      INTEGER          NRRNG(MXRAKE)        ! NUMBER OF RAKE SOLUTION RANGES BINNED INTO DLAMF DEGREE INCREMENTS
      INTEGER          NSOL                 ! NUMBER OF PLANES STORED IN ARRAY SOLNS
      INTEGER          NSRNG(MXSTRK)        ! NUMBER OF STRIKE SOLUTION RANGES BINNED INTO DPHIF DEGREE INCREMENTS
      INTEGER          NSTAR                ! NUMBER OF SOLUTIONS HAVING FIT WITHIN 5% OF FITMIN
      INTEGER          NSTAT                ! TOTAL # OF STATIONS REPORTING FOR ENTIRE DATA SET
      REAL             PHI(MXSTRK)          ! FAULT STRIKE ANGLE IN DEGREES
      REAL             PHIC(MXSTRK)         ! FAULT STRIKE ANGLE IN DEGREES FOR COARSE SEARCH
      REAL             PHIØC                ! INITIAL FAULT STRIKE ANGLE IN DEGREES FOR COARSE SEARCH
      REAL             PHIØF                ! INITIAL FAULT STRIKE ANGLE IN DEGREES FOR FINE SEARCH
      REAL             PHI1F                ! TERMINATING FAULT STRIKE ANGLE IN DEGREES FOR FINE SEARCH
      REAL             POBS(MXSTAT)         ! OBSERVED FIRST MOTION POLARITIES; .5=COMPRESSION, -.5=DILATATION
      REAL             PRAD                 ! RADIATION AMPLITUDE CORRESPONDING AIN, AZ.
                                            ! (DILATATION) -1.<PRAD<+1.(COMPRESSION)
      REAL             PRCNTX               ! % OF STATIONS THAT ARE MACHINE PICKED
      CHARACTER*4      PRMK(MXSTAT)         ! FIRST MOTION DESCRIPTION (EG. IPUØ)
      INTEGER          QCNT(MXQUAL,2)       ! (INPUT) INDX 1=# OF DSCRPNT PLRTIES FOR QLITY, INDX 2=# OF OBSERVATIONS
      REAL             QCNTWT(MXQUAL,2)     ! INDEX 1=WEIGHTED # DSCRPNT POLRITIES FOR QUALITY, INDEX 2=SUM OF WEIGHTS
      REAL             RAD                  ! CONVERSION FROM DEGREES TO RADIANS
      CHARACTER*4      REVSTA(MXSTAT)       ! REVERSED STATION NAMES
      REAL             SA2                  ! SLIP ANGLE OF AUXILLIARY SOLUTION
      INTEGER          SCNT(MXSTAT,2)       ! INDEX 1=# OF DSCRPNT POLARITIES FOR STAT, INDEX 2=# OF OBSERVATIONS
      REAL             SCNTWT(MXSTAT,2)     ! INDEX 1=WEIGHTED # DSCRPNT POLARITIES FOR STAT, INDEX 2=SUM OF WEIGHTS
      CHARACTER*1      SFLAG                ! FLAG INDICATING SECONDARY SOLUTIONS
      REAL             SIGMAF               ! CALCULATED STANDARD DEVIATION OF FIT BASED ON DATA ERRORS
      REAL             SLIP                 ! SLIP ANGLE OF BEST SOLUTION
      REAL             SOLNS(MXSLNS,3)      ! ARRAY OF FINAL SOLUTIONS USED TO CHECK FOR REDUNDANCY
      CHARACTER*4      STAT(MXSTAT)         ! NAMES OF ALL STATIONS REPORTING
      REAL             STDR                 ! STATION DISTIBUTION RATIO
      CHARACTER*4      STN(MXSTAT)          ! STATION NAMES PER EVENT
      CHARACTER*11     STRING               ! SCRATCH VARIABLE
      REAL             STRIKE               ! STRIKE OF BEST SOLUTION
      REAL             SUMWT                ! SUM OF OBSERVED FIRST MOTION WEIGHTS
      CHARACTER*8Ø     TEMP                 ! SCRATCH VARIABLE FOR REFORMATTING EVENT CARD
      CHARACTER*48     TITLE                ! DATA SET DESCRIPTOR
      REAL             TM(6)                ! MOMENT TENSOR IN UPPER TRIANGULAR SYMETRIC STORAGE MODE
      REAL             U(3)                 ! CARTESIAN P-WAVE DIRECTION VECTOR (POSITIVE UP, SOUTH, EAST)
      REAL             WEIGHT(MXQUAL)       ! WEIGHTS ASSOCIATED WITH QUALITIES
      REAL             WTMAX                ! MAXIMUM WEIGHT
```

```
      REAL              WTOBS(MXSTAT)                 ! OBSERVED FIRST MOTIONS WEIGHTS
      REAL              XLAM(MXRAKE)                  ! FAULT RAKE ANGLE IN DEGREES
      REAL              XLAMC(MXRAKE)                 ! FAULT RAKE ANGLE IN DEGREES FOR COARSE SEARCH
      REAL              XLAMØC                        ! INITIAL FAULT RAKE ANGLE IN DEGREES FOR COARSE SEARCH
      REAL              XLAMØF                        ! INITIAL FAULT RAKE ANGLE IN DEGREES FOR FINE SEARCH
      REAL              XLAMIF                        ! TERMINATING FAULT RAKE ANGLE IN DEGREES FOR FINE SEARCH
C
C  INITIALIZE STATISTICS ARRAYS TO ZERO
C
      DATA NSTAT, NFIT /Ø, 2Ø*Ø/
      DATA QCNT, QCNTWT /MXQUAL*Ø, MXQUAL*Ø, MXQUAL*Ø.Ø, MXQUAL*Ø.Ø/
      DATA NDRNG, NSRNG, NRRNG /MXDIP*Ø, MXSTRK*Ø, MXRAKE*Ø/
C
C OPEN FILES
C
      OPEN (UNIT = SUNIT, STATUS = 'NEW', FORM = 'FORMATTED', ACCESS
     & = 'SEQUENTIAL', CARRIAGECONTROL = 'LIST')
      OPEN (UNIT = PUNIT, STATUS = 'NEW', FORM = 'FORMATTED', ACCESS
     & = 'SEQUENTIAL')
      OPEN (UNIT = IUNIT, STATUS = 'OLD', FORM = 'FORMATTED', ACCESS
     & = 'SEQUENTIAL', READONLY, IOSTAT = IERR)
      IF (IERR .NE. Ø) THEN
        PRINT *, '***** FPFIT: ERROR OPENING HYPO FILE; ERROR CODE = ',
     & IERR, ', LOGICAL UNIT # = ', IUNIT, ' *****'
        STOP
      END IF
      OPEN (UNIT = CUNIT, STATUS = 'OLD', FORM = 'FORMATTED', ACCESS
     & = 'SEQUENTIAL', READONLY, IOSTAT = IERR)
      IF (IERR .NE. Ø) THEN
        PRINT *, '***** FPFIT: ERROR OPENING CONTROL FILE; ERROR CODE =
     &', IERR, ', LOGICAL UNIT # = ', CUNIT, ' *****'
        STOP
      END IF
      RAD = ATAN(1.)/45.Ø
      READ (IUNIT, 1Ø)
      READ (IUNIT, 1Ø) TITLE
1Ø    FORMAT (4ØX, A48)
      WRITE (PUNIT, *) TITLE
      WRITE (EUNIT, *) TITLE
C
C READ IN CONTROL PARAMETERS
C
      IF (.NOT. FPINP (CUNIT, DDELC, DDELF, DELØC, DFITC, DISTMX,
     & DLAMC, DLAMF, DPHIC, DPHIF, ERATE, EUNIT, FMAGMN, IJEFF, IPRNT,
     & IRES, IUNIT, KILSTA, MINOBS, MXDIP, MXQUAL, MXRAKE, MXSTAT,
     & MXSTRK, NDELC, NDELFDF, NEV, NKIL, NLAMC, NLAMFDF, NPHIC,
     & NPHIFDF, NREV, PHIØC, REVSTA, WEIGHT, XLAMØC)) STOP
      IF (IPRNT .EQ. 1) OPEN (UNIT = FUNIT, STATUS = 'NEW', FORM =
     & 'FORMATTED', ACCESS = 'SEQUENTIAL')
C
C      WRITE (SUNIT, 11) ' DATE    ORIGIN    LAT N    LONG W     DEPTH
C     1MAG NO GAP   DM  RMS  ERH  ERZ QL  DD DA  SA   FIT  NR  SMWT  BOT
C     2   XX S,D,R RNG  Q'
C11    FORMAT(A)
C
      IEVP = Ø
      IEVR = Ø
C
C FIND MAXIMUM WEIGHT
C
      WTMAX = Ø.
      DO 15 I = 1, MXQUAL
        IF (WEIGHT(I) .GT. WTMAX) WTMAX = WEIGHT(I)
15    CONTINUE
```

```
C
C READ NEXT EVENT
C
2Ø      IEVR = IEVR + 1
        CALL READEQ (AIN, AZ, DIST, DISTMX, EUNIT, EVENT, FMAGMN,
     & IJEFF, IUNIT, KILSTA, MINOBS, MXQUAL, MXSTAT, NKIL, NR, NREV,
     & POBS, PRCNTX, PRMK, REVSTA, SIGMAF, STN, SUMWT, WEIGHT, WTOBS)
        IF (NR .EQ. -1 .OR. IEVP .EQ. NEV) THEN
C
C END OF DATA, CLOSE FILES
C
          IEVR = IEVR - 1
          CLOSE (CUNIT)
          CLOSE (IUNIT)
          CLOSE (PUNIT)
          CLOSE (SUNIT)
          IF (IPRNT .EQ. 1) CLOSE (FUNIT)
          IF (NSTAT .GT. Ø) THEN
            CALL FPOUT (DDELF, DLAMF, DPHIF, ERATE, EUNIT, IEVP, IEVR,
     & IND, IRES, MXDIP, MXQUAL, MXRAKE, MXSTAT, MXSTRK, NDELF, NDRNG,
     & NFIT, NLAMF, NPHIF, NREV, NRRNG, NSRNG, NSTAT, QCNT, QCNTWT,
     & REVSTA, SCNT, SCNTWT, STAT)
C
          ELSE
            WRITE (EUNIT, *) '***** FPFIT ERROR: NO EVENTS SATISFY INPUT
     & CRITERIA *****'
          END IF
          CLOSE (EUNIT)
          STOP
        END IF
C
C INSUFFICIENT READINGS SKIP EVENT
C
        IF (NR .EQ. Ø) GOTO 2Ø
C
C GOOD EVENT: BEGIN EVENT LOOP
C SET UP P-WAVE DIRECTION UNIT VECTOR  (UP, SOUTH, EAST) FOR EACH RAY
C
        DO 3Ø I = 1, NR
          AINR = AIN(I)*RAD
          AZR = AZ(I)*RAD
          U(1) = -COS(AINR)
          U(2) = -SIN(AINR)*COS(AZR)
          U(3) = SIN(AINR)*SIN(AZR)
C
C FIND EXCITATION COEFFICIENTS FOR DETERMINING FAR-FIELD P RADIATION PATTERN
C
          CALL PEXCF (COEF, I, MXSTAT, U)
3Ø      CONTINUE
C
C COARSE LOOP THROUGH FAULT MODELS
C
        IEVP = IEVP + 1
        FIRST = .TRUE.
        CALL SEARCH (BOT, COEF, DDELC, DEL, DELC, DELØC, DFITC, DLAMC,
     & DPHIC, FIRST, FIT, FITMIN, FLAG, GFIT, IGOOD, IPRNT, J1, M1,
     & MXDIP, MXRAKE, MXSTAT, MXSTRK, N1, NDELC, NG, NLAMC, NPHIC, NR,
     & NSTAR, PHI, PHIC, PHIØC, POBS, RAD, WTOBS, XLAM, XLAMC, XLAMØC)
C
C DETERMINE DISTINCT SOLUTIONS FROM ARRAY OF "GOOD" SOLUTIONS IN COARSE SEARCH
C
        CALL HHOG (EUNIT, J1, N1, M1, IGOOD, GFIT, NG, IDST, NDST,
     & MXDIP, MXSLNS, MXSTRK, MXRAKE, NDELC, NPHIC, NLAMC)
C
```

```
C BEGIN LOOP FOR FINE SEARCH CENTERED ON EACH DISTINCT SOLUTION FOUND BY HHOG
C
      FIT9Ø = 1.282 * SIGMAF
      NSOL = Ø
      AERR = 2.*AMAX1(DDELF, DLAMF, DPHIF)
      DO 27Ø ID = 1, NDST
        JI = IDST(ID, 1)
        NI = IDST(ID, 2)
        MI = IDST(ID, 3)
        BEST = .FALSE.
        IF (ID .EQ. 1) BEST = .TRUE.
        FIRST = .FALSE.
C
C  DETERMINE FINE SEARCH RANGE FOR UNRESTRICTED SEARCH
C
        IF (IRES .EQ. Ø) THEN
          NDELF = NDELFDF
          NPHIF = NPHIFDF
          NLAMF = NLAMFDF
          DELØF = DELC(J1) - FLOAT(NDELF/2)*DDELF
          PHIØF = PHIC(N1) - FLOAT(NPHIF/2)*DPHIF
          XLAMØF = XLAMC(M1) - FLOAT(NLAMF/2)*DLAMF
        ELSE
C
C  DETERMINE FINE SEARCH DIP RANGE FOR RESTRICTED SEARCH
C
          DELØF = DELC(J1) - FLOAT(NDELC/2)*DDELC
          IF (DELØF .LT. DELØC) DELØF = DELØC
          IF (DELØF .LT. Ø.) DELØF = Ø.
          DELIF = DELC(J1) + FLOAT(NDELC/2)*DDELC
          IF (DELIF .GT. DELC(NDELC)) DELIF = DELC(NDELC)
          IF (DELIF .GT. 9Ø.) DELIF = 9Ø.
          NDELF = INT((DELIF - DELØF)/DDELF) + 1
          IF (NDELF .GT. MXDIP) THEN
            NDELF = MXDIP
            DELØF = DELC(J1) - FLOAT(NDELF/2)*DDELF
4Ø          IF (DELØF + FLOAT(NDELF - 1)*DDELF .GT. DELC(NDELC)) THEN
              NDELF = NDELF - 1
              GOTO 4Ø
            END IF
          END IF
C
C  DETERMINE FINE SEARCH STRIKE RANGE FOR RESTRICTED SEARCH
C
          PHIØF = PHIC(N1) - FLOAT(NPHIC/2)*DPHIC
          IF (PHIØF .LT. PHIØC) PHIØF = PHIØC
          PHI1F = PHIC(N1) + FLOAT(NPHIC/2)*DPHIC
          IF (PHI1F .GT. PHIC(NPHIC)) PHI1F = PHIC(NPHIC)
          NPHIF = INT((PHI1F - PHIØF)/DPHIF) + 1
          IF (NPHIF .GT. MXSTRK) THEN
            NPHIF = MXSTRK
            PHIØF = PHIC(N1) - FLOAT(NPHIF/2)*DPHIF
            IF (PHIØF .LT. PHIØC) PHIØF = PHIØC
5Ø          IF (PHIØF + FLOAT(NPHIF - 1)*DPHIF .GT. PHIC(NPHIC)) THEN
              NPHIF = NPHIF - 1
              GOTO 5Ø
            END IF
          END IF
C
C  DETERMINE FINE SEARCH RAKE RANGE FOR RESTRICTED SEARCH
C
          XLAMØF = XLAMC(M1) - FLOAT(NLAMC/2)*DLAMC
          IF (XLAMØF .LT. XLAMØC) XLAMØF = XLAMØC
          XLAMIF = XLAMC(M1) + FLOAT(NLAMC/2)*DLAMC
          IF (XLAMIF .GT. XLAMC(NLAMC)) XLAMIF = XLAMC(NLAMC)
```

```
              NLAMF = INT((XLAMIF - XLAMØF)/DLAMF) + 1
              IF (NLAMF .GT. MXRAKE) THEN
                NLAMF = MXRAKE
                XLAMØF = XLAMC(M1) - FLOAT(NLAMF/2)*DLAMF
                IF (XLAMØF .LT. XLAMØC) XLAMØF = XLAMØC
6Ø              IF (XLAMØF + FLOAT(NLAMF -1)*DLAMF .GT. XLAMC(NLAMC)) THEN
                  NLAMF = NLAMF - 1
                  GOTO 6Ø
                END IF
              END IF
            END IF
C
C  DO FINE SEARCH
C
            CALL SEARCH (BOT, COEF, DDELF, DEL, DELC, DELØF, FIT9Ø, DLAMF,
     & DPHIF, FIRST, FIT, FITMIN, FLAG, GFIT, IGOOD, IPRNT, J1, M1,
     & MXDIP, MXRAKE, MXSTAT, MXSTRK, N1, NDELF, NG, NLAMF, NPHIF, NR,
     & NSTAR, PHI, PHIC, PHIØF, POBS, RAD, WTOBS, XLAM, XLAMC, XLAMØF)
C
C EXPRESS SOLUTION IN TERMS OF DIP DIRECTION, DIP ANGLE, AND SLIP ANGLE.
C
            CALL REFRMT(DEL(J1), IDIP1, IDPDR1, ISLIP1, PHI(N1), XLAM(M1))
C
C REPLACE EACH PLANE WITH IDIP1 = Ø BY ITS AUXILLIARY PLANE
C
            IF (IDIP1 .EQ. Ø) THEN
              CALL AUXPLN (FLOAT(IDPDR1), FLOAT(IDIP1), FLOAT(ISLIP1),
     & DD2, DA2, SA2)
              IDPDR1 = IFIX(DD2)
              IDIP1  = IFIX(DA2)
              ISLIP1 = IFIX(SA2)
            END IF
C
C FOR CASES WHERE PLANE IS VERTICAL AND RAKE IS -9Ø, FLIP REPRESENTATION TO
C ONE WITH POSITIVE RAKE
C
            IF (IDIP1 .EQ. 9Ø .AND. ISLIP1 .EQ. -9Ø) THEN
              ISLIP1 = ISLIP1 + 18Ø
              IDPDR1 = JMOD( (IDPDR1 + 18Ø), 36Ø)
            END IF
C
C SKIP SOLUTION IF IT IS COMPLIMENTARY (AUXILLIARY PLANE) TO ANOTHER SOLUTION
C OR IF IT IS THE SAME AS ANOTHER SOLUTION IN THE LIST
C
            IF (NSOL .EQ. Ø .OR. (.NOT. COMPL(SOLNS, NSOL, FLOAT(IDPDR1),
     & FLOAT(IDIP1), FLOAT(ISLIP1), AERR, MXSLNS))) THEN
              NSOL = NSOL + 1
              SOLNS(NSOL, 1) = IDPDR1
              SOLNS(NSOL, 2) = IDIP1
              SOLNS(NSOL, 3) = ISLIP1
C
C FIND THE RANGE OF DIP, STRIKE AND RAKE SPANNING EACH GOOD SOLUTION FOR WHICH THE FIT IS .LE. FITLIM
C
              FITLIM = FITMIN + FIT9Ø
              JMAX = Ø
              JMIN = NDELF
              NMAX = Ø
              NMIN = NPHIF
              MMAX = Ø
              MMIN = NLAMF
              DO 7Ø M = 1, NLAMF
                IF (FIT(J1, N1, M) .LE. FITLIM) THEN
                IF (M .LT. MMIN) MMIN = M
                IF (M .GT. MMAX) MMAX = M
              END IF
```

```
7Ø          CONTINUE
            DO 8Ø N = 1, NPHIF
              IF (FIT(J1, N, M1) .LE. FITLIM) THEN
                IF (N .LT. NMIN) NMIN = N
                IF (N .GT. NMAX) NMAX = N
              END IF
8Ø          CONTINUE
            DO 9Ø J = 1, NDELF
              IF (FIT(J, N1, M1) .LE. FITLIM) THEN
                IF (J .LT. JMIN) JMIN = J
                IF (J .GT. JMAX) JMAX = J
              END IF
9Ø          CONTINUE
C
C UPDATE DISTRIBUTION OF DIP, STRIKE, RAKE RANGES FOR BEST SOLUTIONS
C
            IDRNG = MAX(JMAX - J1, J1 - JMIN)*IFIX(DDELF)
            ISRNG = MAX(NMAX - N1, N1 - NMIN)*IFIX(DPHIF)
            IRRNG = MAX(MMAX - M1, M1 - MMIN)*IFIX(DLAMF)
C
C ACCUMULATE STATISTICS ON DISTRIBUTION OF DIP, STRIKE, RAKE RANGES
C FOR BEST SOLUTIONS ONLY
C
            IF (BEST .AND. (IRES .EQ. Ø)) THEN
              INDEX = IDRNG/IFIX(DDELF) + 1
              NDRNG(INDEX) = NDRNG(INDEX) + 1
              INDEX = ISRNG/IFIX(DPHIF) + 1
              NSRNG(INDEX) = NSRNG(INDEX) + 1
              INDEX = IRRNG/IFIX(DLAMF) + 1
              NRRNG(INDEX) = NRRNG(INDEX) + 1
            END IF
C
C ASSIGN QUALITY CODE TO SOLUTION
C
            FTQUAL(2:2) = 'I'
            IF (FITMIN .LE. .Ø25) THEN
              FTQUAL(1:1) = 'A'
            ELSE IF (FITMIN .GT. Ø.1) THEN
              FTQUAL(1:1) = 'C'
            ELSE
              FTQUAL(1:1) = 'B'
            END IF
            IF (ISRNG .LE. 2Ø .AND. IDRNG .LE. 2Ø .AND. IRRNG .LE. 2Ø)
     & THEN
              FTQUAL(3:3) = 'A'
            ELSE IF (ISRNG .GT. 4Ø .AND. IDRNG .GT. 4Ø .AND. IRRNG
     & .GT. 4Ø) THEN
              FTQUAL(3:3) = 'C'
            ELSE
              FTQUAL(3:3) = 'B'
            END IF
C
C WRITE OUT RESULTS
C
            SFLAG = ' '
            IF (.NOT. BEST) SFLAG = '*'
            TEMP = EVENT(1:53)//EVENT(57:6Ø)//EVENT(54:56)//'.Ø'//
     &      EVENT(63:8Ø)
            AVWT = SUMWT/FLOAT(NR)
            STDR = BOT(J1,N1,M1)/SUMWT
            WRITE (EVFIT, 1ØØ) IDPDR1, IDIP1, ISLIP1, FIT(J1, N1, M1),
     &      NR, AVWT, STDR, PRCNTX, ISRNG, IDRNG, IRRNG, FTQUAL, SFLAG
1ØØ         FORMAT (I4, I3, I4, 2X, F4.2, 1X, I3, 1X, 2F5.2, 1X, F4.2,
     &      1X, 3I3, 1X, A3, A1)
```

```
      IF (IPRNT .EQ. 1) WRITE (FUNIT, 11Ø) TEMP, EVFIT
11Ø   FORMAT ('1', A8Ø, A52)
      WRITE (SUNIT, 12Ø) TEMP, EVFIT
12Ø   FORMAT (A8Ø, A52)
      WRITE (PUNIT, 13Ø) TEMP, EVFIT
13Ø   FORMAT (1X, A8Ø, A52)
      WRITE (PUNIT, *) NSTAR
C
C  LOOP OVER SEARCH AREA TO PRINT OUT FIT PARAMETERS
C
      NSTAR = Ø
      DO 2ØØ M = 1, NLAMF
        IF (IPRNT .EQ. 1) THEN
          WRITE (FUNIT, 14Ø) IFIX(XLAM(M))
14Ø       FORMAT ('Ø', //, 4X, 'SLIP ANGLE = ', I4)
          WRITE (FUNIT, 15Ø) (IFIX(PHI(N)), N = 1, NPHIF)
15Ø       FORMAT (' ', ' STRIKE:', 24(2X, I3))
          WRITE (FUNIT, *) '  DIP'
        END IF
        DO 19Ø J = 1, NDELF
          DO 17Ø N = 1, NPHIF
            IF (IPRNT .EQ. 1) THEN
              IFIT(N) = IFIX(1ØØØ.*FIT(J, N, M))
              IF (IFIT(N) .EQ. 1ØØØ) IFIT(N) = 999
            END IF
            IF (FLAG(J, N, M) .EQ. '*') THEN
              NSTAR = NSTAR + 1
              CALL REFRMT(DEL(J), IDIP, IDIPDR, ISLIP, PHI(N),
     & XLAM(M))
              WRITE (STRING, '(I4, I3, I4)') IDIPDR, IDIP, ISLIP
              IF (NSTAR .EQ. 1) THEN
                LINE = STRING
              ELSE
                LINE = LINE(1:(NSTAR - 1)*11)//STRING
              END IF
              IF (NSTAR .EQ. 11) THEN
                NSTAR = Ø
                WRITE (PUNIT, 16Ø) LINE
16Ø             FORMAT (' ', A)
              END IF
            END IF
17Ø       CONTINUE
          IF (IPRNT .EQ. 1) WRITE (FUNIT, 18Ø) IFIX(DEL(J)),
     & (IFIT(N), FLAG(J, N, M), N = 1, NPHIF)
18Ø       FORMAT (' ', I4, 6X, 24(I3, A1, 1X), //)
19Ø     CONTINUE
2ØØ   CONTINUE
      IF (NSTAR .NE. Ø) WRITE (PUNIT, 16Ø) LINE
C
C ACCUMULATE STATISTICS ON FIT PARAMETER DISTRIBUTION FOR BEST SOLUTIONS ONLY
C
      IF (BEST) THEN
        INDEX = IFIX(FIT(J1, N1, M1)/.Ø25) + 1
        IF (INDEX .GT. 2Ø) INDEX = 2Ø
        NFIT(INDEX) = NFIT(INDEX) + 1
      END IF
C
C RECOMPUTE MOMENT TENSOR REPRESENTATION FOR BEST SOLUTION TO CHECK FOR POLARITY DISCREPANCIES
C
      STRIKE = PHI(N1)*RAD
      DIP = DEL(J1)*RAD
      SLIP = XLAM(M1)*RAD
      CALL SHRFLT (STRIKE, DIP, SLIP, TM)
      DO 26Ø K = 1, NR
```

```
              IF (NSTAT .GE. 1) THEN
                DO 21Ø I = 1, NSTAT
                  IF (STN(K) .EQ. STAT(I)) GOTO 22Ø
21Ø           CONTINUE
              END IF
              NSTAT = NSTAT + 1
              IF (NSTAT .GT. MXSTAT) THEN
                WRITE (EUNIT, *) '***** FPFIT ERROR: # OF STATIONS HAVE
     &POLARITY DISCEPANCIES EXCEEDS ', MXSTAT, ' *****'
                WRITE (EUNIT, *) ' STATION ', STN(K), ' NOT REPORTED IN
     &FINAL SUMMARY'
                GOTO 24Ø
              END IF
              I = NSTAT
              STAT(NSTAT) = STN(K)
              SCNT(NSTAT, 1) = Ø
              SCNT(NSTAT, 2) = Ø
              SCNTWT(NSTAT, 1) = Ø.
              SCNTWT(NSTAT, 2) = Ø.
22Ø           READ (PRMK(K), '(3X, I1)') IPWT
              IF (PRMK(K)(1:1) .EQ. 'X' .OR. PRMK(K)(1:1) .EQ. 'Y')
     & IPWT = IPWT + MXQUAL/2
C
C RECOMPUTE RADIATION PATTERN
C
              PRAD = Ø
              DO 23Ø L = 1, 6
                PRAD = PRAD + TM(L)*COEF(K, L)
23Ø           CONTINUE
C
C CHECK POLARITY AND UPDATE APPROPRIATE STATION COUNT
C
              IF (SIGN(Ø.5, PRAD) .NE. POBS(K)) THEN
                IF (BEST) THEN
                  SCNT(I, 1) = SCNT(I, 1) + 1
                  SCNTWT(I, 1) = SCNTWT(I, 1) +
     & WTOBS(K)*SQRT(ABS(PRAD))
                  QCNT(IPWT + 1, 1) = QCNT(IPWT + 1, 1) + 1
                  QCNTWT(IPWT + 1, 1) = QCNTWT(IPWT + 1, 1) +
     & WTOBS(K)*SQRT(ABS(PRAD))
                END IF
                BDFLAG = '*'
              ELSE
                BDFLAG = ' '
              END IF
              IF (BEST) THEN
                SCNT(I, 2) = SCNT(I, 2) + 1
                SCNTWT(I, 2) = SCNTWT(I, 2) +
     & WTOBS(K)*SQRT(ABS(PRAD))
                QCNT(IPWT + 1, 2) = QCNT(IPWT + 1, 2) + 1
                QCNTWT(IPWT + 1, 2) = QCNTWT(IPWT + 1, 2) +
     & WTOBS(K)*SQRT(ABS(PRAD))
              END IF
C
C WRITE OUT TO POLARITY FILE
C
24Ø           WRITE (PUNIT, 25Ø) STN(K), DIST(K), AZ(K), AIN(K),
     & PRMK(K), WTOBS(K)/WTMAX, BDFLAG
25Ø           FORMAT (1X, A4, 3F6.1, 3X, A4, F5.2, 2X, A1)
26Ø         CONTINUE
            WRITE (PUNIT, *) '                    '
          END IF
C
C END OF SOLUTION LOOP
```

```
C
27Ø      CONTINUE
         IF (NSOL .GT. 1) WRITE (EUNIT, *) '***** EVENT ', EVENT(1:17),
     & ' HAS MULTIPLE SOLUTIONS  *****'
C
C END OF EVENT
C
      GOTO 2Ø
      END
C
C
C
C
C
      LOGICAL FUNCTION FPINP (CUNIT, DDELC, DDELF, DELØC, DFITC, DISTMX,
     & DLAMC, DLAMF, DPHIC, DPHIF, ERATE, EUNIT, FMAGMN, IJEFF, IPRNT,
     & IRES, IUNIT, KILSTA, MINOBS, MXDIP, MXQUAL, MXRAKE, MXSTAT,
     & MXSTRK, NDELC, NDELF, NEV, NKIL, NLAMC, NLAMF, NPHIC, NPHIF,
     & NREV, PHIØC, REVSTA, WEIGHT, XLAMØC)
C
C READ IN CONTROL PARAMETERS
C
      INTEGER          CUNIT                      ! (INPUT) LOGICAL UNIT # OF INPUT CONTROL FILE
      REAL             DDELC                      ! (OUTPUT) FAULT DIP INCREMENT IN DEGREES FOR COARSE SEARCH
      REAL             DDELF                      ! (OUTPUT) FAULT DIP INCREMENT IN DEGREES FOR FINE SEARCH
      REAL             DELØC                      ! (OUTPUT) INITIAL FAULT DIP ANGLE IN DEGREES FOR COARSE SEARCH
      REAL             DFITC                      ! (OUTPUT) INCREMENT TO COARSE FIT FUNCTION
      REAL             DLAMC                      ! (OUTPUT) FAULT RAKE INCREMENT IN DEGREES FOR COARSE SEARCH
      REAL             DLAMF                      ! (OUTPUT) FAULT RAKE INCREMENT IN DEGREES FOR FINE SEARCH
      REAL             DPHIC                      ! (OUTPUT) FAULT STRIKE INCREMENT IN DEGREES FOR COARSE SEARCH
      REAL             DPHIF                      ! (OUTPUT) FAULT STRIKE INCREMENT IN DEGREES FOR FINE SEARCH
      REAL             ERATE(MXQUAL)              ! (OUTPUT) ASSUMED WEIGHTED ERROR RATES FOR DATA, READ FROM CONTROL CARD
      INTEGER          EUNIT                      ! (INPUT) LOGICAL UNIT # OF OUTPUT OF ERROR MESSAGES
      REAL             FMAGMN                     ! (OUTPUT) MINIMUM PERMITTED MAGNITUDE
      INTEGER          IJEFF                      ! (OUTPUT) FLAG: 1(Ø)=DO (NOT) USE DATA WEIGHTED OUT BY JEFFREY'S WEIGHTING
      INTEGER          IPRNT                      ! (OUTPUT) FLAG: 1(Ø)=DO (NOT) PRINT OUT FIT PARAMETERS
      INTEGER          IRES                       ! (OUTPUT) FLAG: Ø(1)=(UN)RESTRICTED SEARCH
      CHARACTER*4      KILSTA(MXSTAT)             ! (OUTPUT) IGNORED STATION NAMES
      INTEGER          MINOBS                     ! (OUTPUT) MINIMUM NUMBER OF OBSERVATIONS REQUIRED
      INTEGER          MXDIP                      ! (INPUT) MAXIMUM # OF DIP INCREMENTS PERMITTED
      INTEGER          MXQUAL                     ! (INPUT) MAXIMUM # OF QUALITIES PERMITTED
      INTEGER          MXRAKE                     ! (INPUT) MAXIMUM # OF RAKE INCREMENTS PERMITTED
      INTEGER          MXSTAT                     ! (INPUT) MAXIMUM # OF STATIONS PERMITTED
      INTEGER          MXSTRK                     ! (INPUT) MAXIMUM # OF STRIKE INCREMENTS PERMITTED
      INTEGER          NDELC                      ! (OUTPUT) NUMBER OF FAULT DIP INCREMENTS FOR COARSE SEARCH
      INTEGER          NDELF                      ! (OUTPUT) DEFAULT NUMBER OF FAULT DIP INCREMENTS FOR FINE SEARCH
      INTEGER          NEV                        ! (OUTPUT) NUMBER OF EVENTS TO PROCESS
      INTEGER          NKIL                       ! (OUTPUT) NUMBER OF IGNORED STATIONS
      INTEGER          NLAMC                      ! (OUTPUT) NUMBER OF FAULT RAKE INCREMENTS FOR COARSE SEARCH
      INTEGER          NLAMF                      ! (OUTPUT) DEFAULT NUMBER OF FAULT RAKE INCREMENTS FOR FINE SEARCH
      INTEGER          NPHIC                      ! (OUTPUT) NUMBER OF FAULT STRIKE INCREMENTS FOR COARSE SEARCH
      INTEGER          NPHIF                      ! (OUTPUT) DEFAULT NUMBER OF FAULT STRIKE INCREMENTS FOR FINE SEARCH
      INTEGER          NREV                       ! (OUTPUT) NUMBER OF REVERSED STATIONS
      REAL             PHIØC                      ! (OUTPUT) INITIAL FAULT STRIKE ANGLE IN DEGREES FOR COARSE SEARCH
      CHARACTER*4      REVSTA(MXSTAT)             ! (OUTPUT) REVERSED STATION NAMES
      REAL             WEIGHT(MXQUAL)             ! (OUTPUT) WEIGHTS ASSOCIATED WITH QUALITIES
      REAL             XLAMØC                     ! (OUTPUT) INITIAL FAULT RAKE ANGLE IN DEGREES FOR COARSE SEARCH
C
      REAL             DDELCDF                    ! DEFAULT FAULT DIP INCREMENT IN DEGREES FOR COARSE SEARCH
      REAL             DELØCDF                    ! DEFAULT INITIAL FAULT DIP ANGLE IN DEGREES FOR COARSE SEARCH
      REAL             DLAMFDF                    ! DEFAULT FAULT RAKE INCREMENT IN DEGREES FOR FINE SEARCH
      REAL             DPHICDF                    ! DEFAULT FAULT STRIKE INCREMENT IN DEGREES FOR COARSE SEARCH
      CHARACTER*8Ø     FILNAM                     ! NAME OF CONTROL FILE
      INTEGER          I                          ! DUMMY LOOP INDEX
```

```
      INTEGER           ICARD                         ! INPUT LINE #
      INTEGER           INDEX1                        ! POSITION OF ']' IN CONTROL AND DATA FILENAMES
      INTEGER           INDEX2                        ! POSITION OF ';' IN CONTROL AND DATA FILENAMES
      INTEGER           IOS                           ! I/O STATUS SPECIFIER
      INTEGER           NDELCDF                       ! DEFAULT NUMBER OF FAULT DIP INCREMENTS FOR COARSE SEARCH
      INTEGER           NDELFDF                       ! DEFAULT NUMBER OF FAULT DIP INCREMENTS FOR FINE SEARCH
      INTEGER           NLAMCDF                       ! DEFAULT NUMBER OF FAULT RAKE INCREMENTS FOR COARSE SEARCH
      INTEGER           NLAMFDF                       ! DEFAULT NUMBER OF FAULT RAKE INCREMENTS FOR FINE SEARCH
      INTEGER           NPHICDF                       ! DEFAULT NUMBER OF FAULT STRIKE INCREMENTS FOR COARSE SEARCH
      INTEGER           NPHIFDF                       ! DEFAULT NUMBER OF FAULT STRIKE INCREMENTS FOR FINE SEARCH
      REAL              PHIØCDF                       ! DEFAULT INITIAL FAULT STRIKE ANGLE IN DEGREES FOR COARSE SEARCH
      CHARACTER*4       STATN                         ! SCRATCH STATION NAME
      CHARACTER*1       STATUS                        ! FLAG: R=REVERSED, K=IGNORED STATION
      REAL              TMP                           ! SCRATCH VARIABLE
      REAL              XLAMØCDF                      ! DEFAULT INITIAL FAULT RAKE ANGLE IN DEGREES FOR COARSE SEARCH
      REAL              WT                            ! SUM OF WEIGHTS
C
C SET UP DEFAULT GRID SPACING
C
      PARAMETER (PHIØCDF = Ø., DELØCDF = 1Ø., XLAMØCDF = -18Ø.)
      PARAMETER (DPHICDF = 2Ø., DDELCDF = 2Ø., DLAMCDF = 2Ø.)
      PARAMETER (NPHICDF = 9, NDELCDF = 5, NLAMCDF = 18)
      PARAMETER (DPHIFDF = 5., DDELFDF = 5., DLAMFDF = 1Ø.)
      PARAMETER (NPHIFDF = 19, NDELFDF = 19, NLAMFDF = 7)
C
      FPINP = .TRUE.
      INQUIRE (CUNIT, NAME = FILNAM)
      INDEX1 = INDEX(FILNAM, ']') + 1
      INDEX2 = INDEX(FILNAM, ';') + 2
      WRITE (EUNIT, *) 'CONTROL FILE = ', FILNAM(INDEX1:INDEX2)
      INQUIRE (IUNIT, NAME = FILNAM)
      INDEX1 = INDEX(FILNAM, ']') + 1
      INDEX2 = INDEX(FILNAM, ';') + 2
      WRITE (EUNIT, *) 'INPUT DATA FILE = ', FILNAM(INDEX1:INDEX2)
      ICARD = 1
      READ (CUNIT, *, ERR = 1ØØØ) DISTMX, FMAGMN, MINOBS, IPRNT, IJEFF,
     & NEV, DFITC
C
C CHECK PARAMETERS
C
      IF (DISTMX .LE. Ø.) THEN
        WRITE (EUNIT, *) '***** FPINP ERROR: DISTMX .LE. Ø *****'
        FPINP = .FALSE.
      END IF
      IF (IPRNT .NE. Ø .AND. IPRNT .NE. 1) THEN
        WRITE (EUNIT, *) '***** FPINP ERROR: IPRNT .NE. Ø OR 1 *****'
        FPINP = .FALSE.
      END IF
      IF (IJEFF .NE. Ø .AND. IJEFF .NE. 1) THEN
        WRITE (EUNIT, *) '***** FPINP ERROR: IJEFF .NE. Ø OR 1 *****'
        FPINP = .FALSE.
      END IF
      IF (MINOBS .LT. Ø) THEN
        WRITE (EUNIT, *) '***** FPINP ERROR: MINOBS LESS THAN Ø *****'
        FPINP = .FALSE.
      END IF
      IF (NEV .LE. Ø) THEN
        WRITE (EUNIT, *) '***** FPINP ERROR: NEV .LE. THAN Ø *****'
        FPINP = .FALSE.
      END IF
      IF (DFITC .LT. Ø) THEN
        WRITE (EUNIT, *) '***** FPINP ERROR: DFITC .LT. THAN Ø *****'
        FPINP = .FALSE.
      ELSE IF (DFITC .GE.  Ø.25) THEN
```

```
        WRITE (EUNIT, *) '***** WARNING: DFITC TRUNCATED TO Ø.25 *****'
      END IF
C
C READ IN ESTIMATED WEIGHTED ERROR RATES FOR EACH QUALITY CLASS OF HAND-
C PICKED AND MACHINE-PICKED DATA.
C
      IF (.NOT. FPINP) RETURN
      ICARD = ICARD + 1
      READ (CUNIT, *, ERR = 1ØØØ) (ERATE(I), I = 1, MXQUAL/2)
      ICARD = ICARD + 1
      READ (CUNIT, *, ERR = 1ØØØ) (ERATE(I), I = MXQUAL/2 + 1, MXQUAL)
      WT = Ø.
C
C CONVERT ESTIMATED ERROR RATES TO WEIGHTING FACTORS
C
      DO 1Ø I = 1, MXQUAL
        IF (ERATE(I) .LT. Ø.5) THEN
         IF (ERATE(I) .LT. Ø.ØØ1) ERATE(I) = Ø.ØØ1
         WEIGHT(I) = 1./SQRT(ERATE(I) - ERATE(I)*ERATE(I))
        ELSE
         WEIGHT(I) = Ø.Ø
        END IF
        WT = WT + WEIGHT(I)
1Ø    CONTINUE
      IF (WT .EQ. Ø.) THEN
        WRITE (EUNIT, *) '***** FPINP ERROR: ALL WEIGHTS ARE Ø *****'
        FPINP = .FALSE.
      END IF
C
C  READ IN LIST OF REVERSED, IGNORED STATIONS
C
      IF (.NOT. FPINP) RETURN
      NREV = 1
      NKIL = 1
      ICARD = ICARD + 1
2Ø      READ (CUNIT, 3Ø, END = 4Ø, ERR = 1ØØØ) STATUS, STATN
3Ø      FORMAT (A1, 1X, A4)
        IF (STATN .NE. '    ') THEN
          IF (STATUS .EQ. 'R') THEN
            REVSTA(NREV) = STATN
            NREV = NREV + 1
            IF (NREV .GT. MXSTAT) THEN
              WRITE (EUNIT, *) '***** FPINP ERROR: # OF REVERSED STATION
     &S EXCEEDS ', MXSTAT, ' *****'
              FPINP = .FALSE.
            END IF
          ELSE IF (STATUS .EQ. 'K') THEN
            KILSTA(NKIL) = STATN
            NKIL = NKIL + 1
            IF (NKIL .GT. MXSTAT) THEN
              WRITE (EUNIT, *) '***** FPINP ERROR: # OF IGNORED STATIONS
     & EXCEEDS ', MXSTAT, ' *****'
              FPINP = .FALSE.
            END IF
          ELSE
            WRITE (EUNIT, *) '***** FPINP ERROR: STATUS OF STATION ',
     & STATN, ' .NE. "K" OR "R" *****'
            FPINP = .FALSE.
          END IF
          GOTO 2Ø
        END IF
4Ø    NKIL = NKIL - 1
      NREV = NREV - 1
C
```

```
C  READ IN OPTIONAL GRID SEARCH PARAMETERS
C
      IF (.NOT. FPINP) RETURN
      ICARD = ICARD + 1
      READ (CUNIT, *, IOSTAT = IOS) PHIØC, PHI1, DELØC, DEL1, XLAMØC,
     & XLAM1, DPHIC, DDELC, DLAMC, DPHIF, DDELF, DLAMF
C
C  UNRESTRICTED SEARCH AREA
C
      IF (IOS .LT. Ø) THEN
        IRES = Ø
        PHIØC = PHIØCDF
        DPHIC = DPHICDF
        DPHIF = DPHIFDF
        PHI1 = PHIØCDF + (NPHICDF - 1)*DPHICDF
        NPHIC = NPHICDF
        NPHIF = NPHIFDF
        DELØC = DELØCDF
        DDELC = DDELCDF
        DDELF = DDELFDF
        DEL1 = DELØCDF + (NDELCDF - 1)*DDELCDF
        NDELC = NDELCDF
        NDELF = NDELFDF
        XLAMØC = XLAMØCDF
        DLAMC = DLAMCDF
        DLAMF = DLAMFDF
        XLAM1 = XLAMØCDF + (NLAMCDF - 1)*DLAMCDF
        NLAMC = NLAMCDF
        NLAMF = NLAMFDF
C
C  RESTRICED SEARCH AREA
C
      ELSE IF (IOS .EQ. Ø) THEN
        IRES = 1
C
C  CHECK RESTRICTED STRIKE RANGE PARAMETERS FOR CONSISTENCY
C
        IF (PHIØC .GT. PHI1) THEN
          TMP = PHI1
          PHI1 = PHIØC
          PHIØC = TMP
        END IF
        DPHI = PHI1 - PHIØC
        IF ((PHIØC .LT. PHIØCDF) .OR.
     & (PHIØC .GT. PHIØCDF + FLOAT(NPHICDF)*DPHICDF)) THEN
          WRITE (EUNIT, 5Ø) 'PHIØC', PHIØC, PHIØCDF, PHIØCDF +
     & FLOAT(NPHICDF)*DPHICDF
 5Ø       FORMAT (' ', '***** FPINP ERROR: ', A6, ' (= ', F6.1,
     & ') OUTSIDE RANGE OF ', F6.1, ' - ', F6.1, ' *****')
          FPINP = .FALSE.
        END IF
        IF ((PHI1 .LT. PHIØCDF) .OR.
     & (PHI1 .GT. PHIØCDF + FLOAT(NPHICDF)*DPHICDF)) THEN
          WRITE (EUNIT, 5Ø) 'PHI1', PHI1, PHIØCDF, PHIØCDF +
     & FLOAT(NPHICDF)*DPHICDF
          FPINP = .FALSE.
        END IF
        IF (DPHIC .LE. Ø.) THEN
          WRITE (EUNIT, 6Ø) 'DPHIC'
 6Ø       FORMAT (' ', '***** FPINP ERROR: ', A5,
     & ' MUST BE GREATER THAN Ø *****')
          FPINP = .FALSE.
        ELSE
          NPHIC = INT(DPHI/DPHIC) + 1
```

```
          IF (NPHIC .GT. MXSTRK) THEN
            WRITE (EUNIT, 7Ø) 'STRIKE', PHI1, PHIØC, DPHIC, MXSTRK
 7Ø         FORMAT (' ', '***** FPINP ERROR: ', A6,
     & ' RANGE TOO LARGE.  (', F6.1, ' - ', F6.1, ')/', F6.1,
     & ' + 1 GREATER THAN ', I2, ' *****')
            FPINP = .FALSE.
          END IF
        END IF
        IF (DPHIF .LE. Ø.) THEN
          WRITE (EUNIT, 6Ø) 'DPHIF'
          FPINP = .FALSE.
        END IF
C
C  CHECK RESTRICTED DIP RANGE PARAMETERS FOR CONSISTENCY
C
        IF (DELØC .GT. DEL1) THEN
          TMP = DEL1
          DEL1 = DELØC
          DELØC = TMP
        END IF
        DDEL = DEL1 - DELØC
        IF ((DELØC .LT. Ø.) .OR. (DELØC .GT. 9Ø.)) THEN
          WRITE (EUNIT, 5Ø) 'DELØC', DELØC, Ø., 9Ø.
          FPINP = .FALSE.
        END IF
        IF ((DEL1C .LT. Ø.) .OR. (DEL1C .GT. 9Ø.)) THEN
          WRITE (EUNIT, 5Ø) 'DEL1C', DEL1C, Ø., 9Ø.
          FPINP = .FALSE.
        END IF
        IF (DDELC .LE. Ø.) THEN
          WRITE (EUNIT, 6Ø) 'DDELC'
          FPINP = .FALSE.
        ELSE
          NDELC = INT(DDEL/DDELC) + 1
          IF (NDELC .GT. MXSTRK) THEN
            WRITE (EUNIT, 7Ø) 'DIP', DEL1, DELØC, DDELC, MXSTRK
            FPINP = .FALSE.
          END IF
        END IF
        IF (DDELF .LE. Ø.) THEN
          WRITE (EUNIT, 6Ø) 'DDELF'
          FPINP = .FALSE.
        END IF
C
C  CHECK RESTRICTED RAKE RANGE PARAMETERS FOR CONSISTENCY
C
        IF (XLAMØC .GT. XLAM1) THEN
          TMP = XLAM1
          XLAM1 = XLAMØC
          XLAMØC = TMP
        END IF
        DLAM = XLAM1 - XLAMØC
        IF ((XLAMØC .LT. XLAMØCDF) .OR.
     & (XLAMØC .GT. XLAMØCDF + FLOAT(NLAMCDF)*DLAMCDF)) THEN
          WRITE (EUNIT, 5Ø) 'XLAMØC', XLAMØC, XLAMØCDF, XLAMØCDF +
     & FLOAT(NLAMCDF)*DLAMCDF
          FPINP = .FALSE.
        END IF
        IF ((XLAM1 .LT. XLAMØCDF) .OR.
     & (XLAM1 .GT. XLAMØCDF + FLOAT(NLAMCDF)*DLAMCDF)) THEN
          WRITE (EUNIT, 5Ø) 'XLAM1', XLAM1, XLAMØCDF, XLAMØCDF +
     & FLOAT(NLAMCDF)*DLAMCDF
          FPINP = .FALSE.
        END IF
```

```
        IF (DLAMC .LE. Ø.) THEN
          WRITE (EUNIT, 6Ø) 'DLAMC'
          FPINP = .FALSE.
        ELSE
          NLAMC = INT(DLAM/DLAMC) + 1
          IF (NLAMC .GT. MXSTRK) THEN
            WRITE (EUNIT, 7Ø) 'RAKE', XLAMI, XLAMØC, DLAMC, MXSTRK
            FPINP = .FALSE.
          END IF
        END IF
        IF (DLAMF .LE. Ø.) THEN
          WRITE (EUNIT, 6Ø) 'DLAMF'
          FPINP = .FALSE.
        END IF
C
C  READ ERROR
C
      ELSE
        GOTO 1ØØØ
      END IF
C
C WRITE OUT PARAMETERS
C
      IF (FPINP) THEN
        WRITE (EUNIT, *) 'MAXIMUM EPICENTRAL DISTANCE = ', DISTMX
        WRITE (EUNIT, *) 'MINIMUM MAGNITUDE = ', FMAGMN
        WRITE (EUNIT, *) 'MINIMUM # OBSERVATIONS = ', MINOBS
        IF (IPRNT .EQ. 1) THEN
          WRITE (EUNIT, *) 'PARAMETER FIT FILE GENERATED (IPRNT = 1)'
        ELSE
          WRITE (EUNIT, *) 'PARAMETER FIT FILE SUPPRESSED (IPRNT = Ø)'
        END IF
        IF (IJEFF .EQ. 1) THEN
          WRITE (EUNIT, *) 'PHASE DATA REJECTED BY JEFFREYS'' WEIGHTING
     &INCLUDED (IJEFF = 1)'
        ELSE
          WRITE (EUNIT, *) 'PHASE DATA REJECTED BY JEFFREYS'' WEIGHTING
     &EXCLUDED (IJEFF = Ø)'
        END IF
        WRITE (EUNIT, *) 'UPTO ', NEV, '  EVENTS PROCESSED'
        WRITE (EUNIT, *) 'MISFIT RANGE FOR RELATIVE MINIMA IN COARSE SEA
     &RCH = ',DFITC
        WRITE  (EUNIT, 8Ø)
8Ø      FORMAT ('Ø', 'HAND-PICKED DATA:    QUALITY    WEIGHT     WEIGHTE
     &D ERROR RATE (EST.)')
        DO 1ØØ I = 1, MXQUAL/2
          WRITE (EUNIT, 9Ø) I - 1, WEIGHT(I), ERATE(I)
9Ø        FORMAT (' ', 23X, I2, 7X, F6.3, 5X, F6.3)
1ØØ     CONTINUE
        WRITE  (EUNIT, 11Ø)
11Ø     FORMAT ('Ø', 'MACHINE-PICKED DATA: QUALITY    WEIGHT     WEIGHTE
     &D ERROR RATE (EST.)')
        DO 12Ø I = MXQUAL/2 + 1, MXQUAL
          WRITE (EUNIT, 9Ø) I - MXQUAL/2 - 1, WEIGHT(I), ERATE(I)
12Ø     CONTINUE
        IF (NKIL .GT. Ø) THEN
          WRITE (EUNIT, *) 'THE FOLLOWING STATIONS IGNORED IN FAULT-PLAN
     &E CALCULATIONS'
          DO 13Ø I = 1, NKIL
            WRITE (EUNIT, *) KILSTA(I)
13Ø       CONTINUE
        END IF
        IF (IRES .EQ. 1) THEN
          WRITE (EUNIT, 14Ø)
```

```
14Ø       FORMAT ('Ø', 'RESTRICTED SEARCH RANGE:            COARSE SEARCH
     &                  FINE SEARCH')
          WRITE (EUNIT, 15Ø)
15Ø       FORMAT (' ', T28,
     &'START     END   INCRMNT  # INCRMNTS       INCRMNT', /, T28,
     &'-------------------------------------     -------')
          WRITE (EUNIT, 16Ø) 'STRIKE: ', PHIØC, PHI1, DPHIC, NPHIC,DPHIF
16Ø       FORMAT(' ', T19, A8, 3(F6.1, 2X), 9X, I2, 9X, F6.1, 9X, I2)
          WRITE (EUNIT, 16Ø) '   DIP: ', DELØC, DEL1, DDELC, NDELC,DDELF
          WRITE (EUNIT, 16Ø) '  RAKE:  ', XLAMØC, XLAM1, DLAMC, NLAMC,
     & DLAMF
        ELSE
          WRITE (EUNIT, 17Ø)
17Ø       FORMAT ('Ø', 'UNRESTRICTED SEARCH RANGE:          COARSE SEARCH
     &                        FINE SEARCH')
          WRITE (EUNIT, 18Ø)
18Ø       FORMAT (' ', T28,
     &'START     END   INCRMNT  # INCRMNTS       INCRMNT  # INCRMNTS',
     & /, T28,
     & '-------------------------------------     ------------------')
          WRITE (EUNIT, 16Ø) 'STRIKE: ', PHIØC, PHI1, DPHIC, NPHIC,
     & DPHIF, NPHIF
          WRITE (EUNIT, 16Ø) '   DIP: ', DELØC, DEL1, DDELC, NDELC,
     & DDELF, NDELF
          WRITE (EUNIT, 16Ø) '  RAKE:  ', XLAMØC, XLAM1, DLAMC, NLAMC,
     & DLAMF, NLAMF
        END IF
        WRITE (EUNIT, 19Ø)
19Ø     FORMAT ('Ø')
      END IF
      RETURN
C
C READ ERROR
C
1ØØØ   WRITE (EUNIT, 1Ø1Ø) ICARD, CUNIT
1Ø1Ø   FORMAT (' ', '***** FPINP ERROR: READ ERROR ON LINE ', I3,
     & ' ON LOGICAL UNIT ', I2, ' *****')
      FPINP = .FALSE.
      RETURN
C
      END
C
C
C
C
      SUBROUTINE READEQ (AIN, AZ, DIST, DISTMX, EUNIT, EVENT, FMAGMN,
     & IJEFF, IUNIT, KILSTA, MINOBS, MXQUAL, MXSTAT, NKIL, NR, NREV,
     & POBS, PRCNTX, PRMK, REVSTA, SIGMAF, STN, SUMWT, WEIGHT, WTOBS)
C
C READS HYPO71 OUTPUT LISTING. RETURNS SUMMARY CARD AND CORRESPONDING PHASE FIRST MOTIONS, QUALITITES, ANGLES OF INCIDENCE,
C STATION NAMES, AND AZIMUTHS.  CALCULATES STANDARD DEVIATION (SIGMAF) OF FIT FROM ESTIMATED STANDARD DEVIATIONS OF THE DATA.
C THE ESTIMATED DATA ERRORS ARE CONTROL-FILE INPUTS; CORRESPONDING DATA WEIGHTS ARE CALCULATED IN MAIN AND PASSED TO THIS
C ROUTINE IN THE PARAMETER "WEIGHT".
C
      REAL              AIN(MXSTAT)                 ! (OUTPUT) RAY INCIDENCE ANGLES
      REAL              AZ(MXSTAT)                  ! (OUTPUT) RAY AZIMUTH ANGLES (CORRESPONDING TO AIN)
      REAL              DIST(MXSTAT)                ! (OUTPUT) EPICENTRAL DISTANCE
      REAL              DISTMX                      ! (INPUT) MAXIMUM PERMITTED EPICENTRAL DISTANCE
      INTEGER           EUNIT                       ! (INPUT) LOGICAL UNIT # OF OUTPUT OF ERROR MESSAGES
      CHARACTER*8Ø      EVENT                       ! (OUTPUT) SUMMARY CARD
      REAL              FMAGMN                      ! (INPUT) MINIMUM PERMITTED MAGNITUDE
      INTEGER           IJEFF                       ! (INPUT) FLAG: 1(Ø)=DO (NOT) USE DATA WEIGHTED OUT BY JEFFREY'S WEIGHTING
      INTEGER           IUNIT                       ! (INPUT) LOGICAL UNIT # OF HYPO71 LISTING FILE (INPUT FILE)
      CHARACTER*4       KILSTA(MXSTAT)              ! (INPUT) IGNORED STATION NAMES
```

```
      INTEGER           MINOBS                 ! (INPUT) MINIMUM NUMBER OF OBSERVATIONS REQUIRED
      INTEGER           MXQUAL                 ! (INPUT) MAXIMUM # OF QUALITIES PERMITTED
      INTEGER           MXSTAT                 ! (INPUT) MAXIMUM # OF STATIONS PERMITTED
      INTEGER           NKIL                   ! (INPUT) NUMBER OF IGNORED STATIONS
      INTEGER           NR                     ! (OUTPUT) -1=EOF, Ø=SKIP EVENT, NR>Ø => NUMBER OF STATIONS
      INTEGER           NREV                   ! (INPUT) NUMBER OF REVERSED STATIONS
      REAL              POBS(MXSTAT)           ! (OUTPUT) OBSERVED FIRST MOTION POLARITIES; .5=COMPRESSION, -.5=DILATATION
      REAL              PRCNTX                 ! (OUTPUT) X OF STATIONS THAT ARE MACHINE PICKED
      CHARACTER*4       PRMK(MXSTAT)           ! (OUTPUT) FIRST MOTION DESCRIPTION (EG. IPUØ)
      CHARACTER*4       REVSTA(MXSTAT)         ! (INPUT) REVERSED STATION NAMES
      REAL              SIGMAF                 ! (OUTPUT) CALCULATED STANDARD DEVIATION OF FIT BASED ON DATA ERRORS
      CHARACTER*4       STN(MXSTAT)            ! (OUTPUT) STATION NAMES
      REAL              SUMWT                  ! (OUTPUT) SUM OF OBSERVED FIRST MOTION WEIGHTS
      REAL              WEIGHT(MXQUAL)         ! (INPUT) WEIGHTS ASSOCIATED WITH QUALITIES
      REAL              WTOBS(MXSTAT)          ! (INPUT) OBSERVED FIRST MOTIONS WEIGHTS
C
      CHARACTER*1       FM                     ! FIRST MOTION DIRECTION (U, D, +, -)
      REAL              FMAG                   ! EVENT MAGNITUDE
      INTEGER           IPWT                   ! QUALIITY ASSIGNED TO P ARRIVAL
      INTEGER           J                      ! DUMMY LOOP INDEX
      CHARACTER*2       JFRYWT                 ! FLAG: '**' DENOTES P ARRIVAL TIME UNRELIABLE BY JEFFREYS' WEIGHTING
      INTEGER           JWT                    ! INDEX FOR DATA WEIGHT
      INTEGER           K                      ! COUNTER OF GOOD PHASE READINGS
      CHARACTER*8Ø      LINE                   ! LINE OF HYPO71 STATION OUTPUT
      CHARACTER*1       M                      ! TEST FOR FORTRAN CARRIAGE CONTROL
      INTEGER           NCLAS(2Ø)              ! NUMBER OF OBSERVATIONS IN EACH DATA CLASS
      CHARACTER*4       TEST                   ! 2ND-6TH CHARACTERS OF LINE OF HYPO71 OUTPUT
      REAL              VARF                   ! CALCULATED VARIANCE IN FIT BASED ON DATA ERRORS.
      REAL              WT                     ! WEIGHT ASSIGNED TO P ARRIVAL
C
C RESET VALUES
C
      DO 1Ø I = 1, MXQUAL
        NCLAS(I) = Ø
1Ø    CONTINUE
C
C FIND LINE PRIOR TO SUMMARY CARD
C
      PRCNTX = Ø.
      SUMWT = Ø.
2Ø    READ (IUNIT, 3Ø, END = 1ØØØ) TEST
3Ø    FORMAT (2X, A4)
      IF (TEST .NE. 'DATE') GOTO 2Ø
C
C READ SUMMARY CARD
C
        READ (IUNIT, 4Ø, END = 1ØØØ) EVENT
4Ø      FORMAT (1X, A8Ø)
        READ (EVENT, '(45X, F5.2)') FMAG
C
C CHECK MAGNTITUDE
C
        IF (FMAG .LT. FMAGMN) THEN
          NR = Ø
          RETURN
        END IF
        READ (IUNIT, 3Ø, END = 1ØØØ) TEST
        READ (IUNIT, 3Ø, END = 1ØØØ) TEST
C
C CHECK WHETHER PHASE DATA OR FOCAL MECHANISM FOLLOW
C
        IF (TEST .NE. 'STN ') GOTO 2Ø
        K = I
5Ø      STN(K) = '    '
```

```
      READ (1UNIT, 6Ø, END = 7Ø) LINE
6Ø    FORMAT (A)
      READ (LINE, '(A1, A4)') M, STN(K)
C
C CHECK FOR END OF PHASE DATA
C
7Ø    IF (M .EQ. '1' .OR. STN(K) .EQ. '    ') THEN
C
C END OF EVENT
C
        IF (K - 1 .GE. MINOBS) THEN
          NR = K - 1
          PRCNTX = PRCNTX/FLOAT(NR)
          VARF1 = Ø.
          VARF2 = Ø.
          DO 8Ø JWT = 1, MXQUAL
            IF (WEIGHT(JWT) .GE. .ØØØ1) THEN
              VARF1 = VARF1 + NCLAS(JWT)
              VARF2 = VARF2 + NCLAS(JWT)*WEIGHT(JWT)
            END IF
8Ø        CONTINUE
          VARF  = VARF1/(VARF2*VARF2)
          SIGMAF= SQRT(VARF)
        ELSE
          NR = Ø
        END IF
        RETURN
      END IF
C
C CHECK FOR REPEATED PHASE CARD
C
      IF (K .GT. 2) THEN
        DO 1ØØ J = 1, K - 1
          IF (STN(K) .EQ. STN(J)) THEN
            WRITE (EUNIT, 9Ø) STN(K), EVENT(1:18)
9Ø          FORMAT (' ', '***** READEQ ERROR: ', A4,
     & ' HAS MULTIPLE PHASE CARDS FOR EVENT:', A18, ' *****')
            GOTO 5Ø
          END IF
1ØØ     CONTINUE
      END IF
C
C IGNORE THIS STATION?
C
      IF (NKIL .GT. Ø) THEN
        DO 11Ø I = 1, NKIL
          IF (STN(K) .EQ. KILSTA(I)) GOTO 5Ø
11Ø     CONTINUE
      END IF
C
C  SO FAR, SO GOOD: NOW CHECK PHASE CARD FOR POLARITY, DISTANCE, JEFFREYS' WEIGHTING, QUALITY
C
      READ (LINE, 12Ø) DIST(K), AZ(K), AIN(K), PRMK(K), JFRYWT
12Ø   FORMAT (6X, F5.1, 1X, F3.Ø, 1X, F3.Ø, 1X, A4, 35X, A2)
      READ (PRMK(K), '(2X, A1, I1)') FM, IPWT
      IF (FM .NE. 'U' .AND. FM .NE. 'D' .AND. FM .NE. '+' .AND.
     & FM .NE. '-') GOTO 5Ø
      IF (DIST(K) .GT. DISTMX) GOTO 5Ø
      IF (IJEFF .EQ. Ø .AND. JFRYWT .EQ. '**') GOTO 5Ø
      IF (IPWT .GE. MXQUAL/2) THEN
        WT = Ø.
      ELSE IF (PRMK(K)(1:1) .EQ. 'X' .OR. PRMK(K)(1:1) .EQ. 'Y') THEN
        JWT = IPWT + MXQUAL/2 + 1
        WT = WEIGHT(JWT)
```

```
          IF (WT .NE. Ø.) PRCNTX = PRCNTX + 1.
        ELSE
          JWT = IPWT + 1
          WT = WEIGHT(JWT)
        END IF
        IF (WT .EQ. Ø.) GOTO 5Ø
C
C FLIP POLARIITES IF STATION IS DESIGNATED AS REVERSED
C
        DO 13Ø I = 1, NREV
          IF (STN(K) .EQ. REVSTA(I)) THEN
            IF (FM .EQ. 'U') PRMK(K)(3:3) = 'D'
            IF (FM .EQ. 'D') PRMK(K)(3:3) = 'U'
            IF (FM .EQ. '+') PRMK(K)(3:3) = '-'
            IF (FM .EQ. '-') PRMK(K)(3:3) = '+'
            FM = PRMK(K)(3:3)
          END IF
13Ø     CONTINUE
C
        NCLAS(JWT) = NCLAS(JWT) + 1
        WTOBS(K) = WT
        SUMWT = SUMWT + WT
        IF (FM .EQ. 'U' .OR. FM .EQ. '+') THEN
          POBS(K) = .5
        ELSE
          POBS(K) = -.5
        END IF
C
C INCREMENT K AND CHECK NUMBER AGAINST ARRAY DIMENSIONS
C
        K = K + 1
        IF (K .GT. MXSTAT) THEN
          WRITE (EUNIT, *) '***** READEQ ERROR: NUMBER OF STATIONS READI
     &NGS EXCEEDS ', MXSTAT, 'FOR EVENT:', EVENT(1:18), ' *****'
          IF (NR .GT. MINOBS) THEN
            NR = K - 1
            PRCNTX = PRCNTX/FLOAT(NR)
          ELSE
            NR = Ø
          END IF
          RETURN
        END IF
C
C READ ANOTHER PHASE
C
        GOTO 5Ø
C
C END OF FILE
C
1ØØØ  NR = -1
      RETURN
      END
C
C
C
C
C
      SUBROUTINE FPOUT (DDELF, DLAMF, DPHIF, ERATE, EUNIT, IEVP, IEVR,
     & IND, IRES, MXDIP, MXQUAL, MXRAKE, MXSTAT, MXSTRK, NDELF, NDRNG,
     & NFIT, NLAMF, NPHIF, NREV, NRRNG, NSRNG, NSTAT, QCNT, QCNTWT,
     & REVSTA, SCNT, SCNTWT, STAT)
C
C GENERATE SUMMARY LISTING OF POLARITY DISCREPANCIES AS A FUNCTION OF STATION AND QUALITY, THE DISTRIBUTION OF
C FIT PARAMETERS, AND THE DISTRIBUTION OF DIP, STRIKE, AND RAKE RANGES ABOUT BEST FIT SOLUTION
```

```
C
      REAL            DDELF              ! (INPUT) FAULT DIP INCREMENT IN DEGREES FOR FINE SEARCH
      REAL            DLAMF              ! (INPUT) FAULT RAKE INCREMENT IN DEGREES FOR FINE SEARCH
      REAL            DPHIF              ! (INPUT) FAULT STRIKE INCREMENT IN DEGREES FOR FINE SEARCH
      REAL            ERATE(MXQUAL)      ! (INPUT) ESTIMATED WEIGHTED ERROR RATES
      INTEGER         EUNIT              ! (INPUT) LOGICAL UNIT # OF OUTPUT OF ERROR MESSAGES
      INTEGER         IEVP               ! (INPUT) # OF EVENTS PROCESSED
      INTEGER         IEVR               ! (INPUT) # OF EVENTS READ
      INTEGER         IND(MXSTAT)        ! (INPUT) POINTER ARRAY TO SORTED ORDER
      INTEGER         IRES               ! (INPUT) FLAG: Ø(1)=(UN)RESTRICTED SEARCH
      INTEGER         MXDIP              ! (INPUT) MAXIMUM # OF DIP INCREMENTS PERMITTED
      INTEGER         MXQUAL             ! (INPUT) MAXIMUM # OF QUALITIES PERMITTED
      INTEGER         MXRAKE             ! (INPUT) MAXIMUM # OF RAKE INCREMENTS PERMITTED
      INTEGER         MXSTAT             ! (INPUT) MAXIMUM # OF STATIONS PERMITTED
      INTEGER         MXSTRK             ! (INPUT) MAXIMUM # OF STRIKE INCREMENTS PERMITTED
      INTEGER         NDELF              ! (INPUT) # OF FAULT DIP INCREMENTS FOR FINE SEARCH
      INTEGER         NDRNG(MXDIP)       ! (INPUT) # OF DIP SOLUTION RANGES BINNED INTO DDELF DEGREE INCREMENTS
      INTEGER         NFIT(2Ø)           ! (INPUT) # OF SOLUTIONS BINNED INTO .Ø25 FIT INCREMENTS
      INTEGER         NLAMF              ! (INPUT) # OF FAULT RAKE INCREMENTS FOR FINE SEARCH
      INTEGER         NPHIF              ! (INPUT) # OF FAULT STRIKE INCREMENTS FOR FINE SEARCH
      INTEGER         NREV               ! (INPUT) # OF REVERSED STATIONS
      INTEGER         NRRNG(MXRAKE)      ! (INPUT) # OF RAKE SOLUTION RANGES BINNED INTO DLAMF DEGREE INCREMENTS
      INTEGER         NSRNG(MXSTRK)      ! (INPUT) # OF STRIKE SOLUTION RANGES BINNED INTO DPHIF DEGREE INCREMENTS
      INTEGER         NSTAT              ! (INPUT) TOTAL # OF STATIONS REPORTING FOR ENTIRE DATA SET
      INTEGER         QCNT(MXQUAL,2)     ! (INPUT) INDX 1=# OF DSCRPNT PLRTIES FOR QLITY, INDX 2=# OF OBSERVATIONS
      REAL            QCNTWT(MXQUAL,2)   ! (INPUT) INDX 1=WEIGHTED # DSCRPNT PLRTIES FOR QLITY, INDX 2=SUM OF WEIGHTS
      CHARACTER*4     REVSTA(MXSTAT)     ! (INPUT) REVERSED STATION NAMES
      INTEGER         SCNT(MXSTAT,2)     ! (INPUT) INDEX 1=# OF DSCRPNT POLARITIES FOR STAT, INDEX 2=# OF OBSRVATIONS
      REAL            SCNTWT(MXSTAT,2)   ! (INPUT) INDX 1=WEIGHTED # DSCRPNT POLRTIES FOR STAT, INDX 2=SUM OF WEIGHTS
      CHARACTER*4     STAT(MXSTAT)       ! (INPUT) NAMES OF ALL STATIONS REPORTING
C
      CHARACTER*2     ESTAR              ! FLAG INDICATES LARGE DISCREPANCY BETWEEN ACTUAL AND ESTIMATED ERROR RATES
      INTEGER         I                  ! LOOP INDEX
      INTEGER         J                  ! DUMMY VARIABLE
      INTEGER         NATOT              ! SUM OF # OF POLARITIES IN AGREEMENT WITH SOLUTION
      INTEGER         NDTOT              ! SUM OF # OF POLARITIES IN DISCREPANCY WITH SOLUTION
      INTEGER         NTOT               ! TOTAL # OF OBSERVATIONS
      REAL            RATE               ! WEIGHTED ERROR RATE PER QUALITY CLASS
      CHARACTER*1     STAR               ! DENOTES STATION DESIGNATED AS REVERSED
      REAL            WTOT               ! SUMMATION OF WEIGHTS OVER ALL STATIONS
C
      NDTOT = Ø
      NTOT = Ø
      WTOT = Ø.
      DO 5 I = 1, NSTAT
        NDTOT = NDTOT + SCNT(I, 1)
        NTOT = NTOT + SCNT(I, 2)
        WTOT = WTOT + SCNTWT(I, 2)
5     CONTINUE
      NATOT = NTOT - NDTOT
C
C WRITE OUT SUMMARY OF POLARITY DISCREPANCIES BY STATION
C
      WRITE (EUNIT, 1Ø)
1Ø    FORMAT ('Ø', 'SUMMARY OF STATIONS HAVING POLARITIES IN DISCREPANCY
     & WITH BEST FIT SOLUTION (* DENOTES REVERSED STATION)', /,
     & ' STATION     DISCREPANCIES    AGREEMENTS       TOTAL
     &WEIGHTED ERROR RATE   TOTAL ERROR CONTRIBUTION', /)
C
C SORT STATIONS ALPHABETICALLY
C
      CALL CSORT (STAT, IND, NSTAT)
      DO 4Ø I = 1, NSTAT
        J = IND(I)
```

```
          STAR = ' '
          DO 2Ø K = 1, NREV
            IF (STAT(J) .EQ. REVSTA(K)) STAR = '*'
2Ø        CONTINUE
          WRITE (EUNIT, 3Ø) STAR, STAT(J), SCNT(J, 1), SCNT(J, 2) -
     & SCNT(J, 1), SCNT(J, 2), SCNTWT(J, 1)/SCNTWT(J, 2),
     & SCNTWT(J, 1)/WTOT
3Ø        FORMAT (' ', A1, A4, 3(1ØX, I5), 9X, F6.3, 1ØX, F6.4)
4Ø      CONTINUE
        WRITE (EUNIT, 5Ø) NDTOT, NATOT, NTOT
5Ø      FORMAT ('Ø', 'TOTAL', 3(1ØX, I5))
C
C WRITE OUT SUMMARY OF HAND-PICKED POLARITY DISCREPANCIES BY READING QUALITY
C
        WRITE (EUNIT, 7Ø)
7Ø      FORMAT ('Ø', 'SUMMARY OF HAND-PICKED DATA WITH RESPECT TO
     & BEST FIT SOLUTIONS', /,
     & ' QUAL          DISCREPANCIES    AGREEMENTS       TOTAL
     &   WEIGHTED ERROR RATE', /)
        NDTOT = Ø
        NTOT = Ø
        DO 9Ø I = 1, MXQUAL/2
          ESTAR = '  '
          NDTOT = NDTOT + QCNT(I, 1)
          NTOT = NTOT + QCNT(I, 2)
          IF (QCNTWT(I, 2) .EQ. Ø.) THEN
            RATE = Ø.
          ELSE
            RATE = QCNTWT(I, 1)/QCNTWT(I, 2)
            IF (RATE .GE. Ø.ØØØ1) THEN
              IF (ABS((ERATE(I)-RATE)/RATE) .GE. Ø.2) ESTAR = '**'
            END IF
          END IF
          WRITE (EUNIT, 8Ø) I - 1, QCNT(I, 1), QCNT(I, 2) -
     & QCNT(I, 1), QCNT(I, 2), RATE, ESTAR
8Ø        FORMAT (' ', 2X, I1, 2X, 3(1ØX, I5), 9X, F6.4, 1X, A3)
9Ø      CONTINUE
        NATOT = NTOT - NDTOT
        WRITE (EUNIT, 5Ø) NDTOT, NATOT, NTOT
C
C WRITE OUT SUMMARY OF MACHINE-PICKED POLARITY DISCREPANCIES BY READING QUALITY
C
        WRITE (EUNIT, 11Ø)
11Ø     FORMAT ('Ø', 'SUMMARY OF MACHINE-PICKED DATA WITH RESPECT TO
     & BEST FIT SOLUTIONS', /,
     & ' QUAL          DISCREPANCIES    AGREEMENTS       TOTAL
     &   WEIGHTED ERROR RATE', /)
        NDTOT = Ø
        NTOT = Ø
        DO 12Ø I = MXQUAL/2 + 1, MXQUAL
          ESTAR =  '  '
          NDTOT = NDTOT + QCNT(I, 1)
          NTOT = NTOT + QCNT(1, 2)
          IF (QCNTWT(I, 2) .EQ. Ø.) THEN
            RATE = Ø.
          ELSE
            RATE = QCNTWT(I, 1)/QCNTWT(I, 2)
            IF (RATE .GE. Ø.ØØØ1) THEN
              IF (ABS((ERATE(I)-RATE)/RATE) .GE. Ø.2)  ESTAR = '**'
            END IF
          END IF
          WRITE (EUNIT, 8Ø) I - MXQUAL/2 - 1, QCNT(I, 1), QCNT(I, 2) -
     & QCNT(I, 1), QCNT(I, 2), RATE, ESTAR
12Ø     CONTINUE
```

```
      NATOT = NTOT - NDTOT
      WRITE (EUNIT, 5Ø) NDTOT, NATOT, NTOT
C
C WRITE OUT DISTRIBUTION OF FIT PARAMETERS
C
      WRITE (EUNIT, 13Ø)
13Ø   FORMAT ('Ø', 'DISTRIBUTION OF SOLUTION MISFIT SCORES', /,
     & 1X, ' MISFIT SCORE        #')
      DO 15Ø I = 1, 2Ø
        WRITE (EUNIT, 14Ø) FLOAT(I - 1)*.Ø25, FLOAT(I)*.Ø25, NFIT(I)
14Ø     FORMAT (' ', F5.3, ' - ', F5.3, 3X, I5)
15Ø   CONTINUE
C
C WRITE OUT DISTRIBUTION OF DIP, STRIKE, RAKE RANGES FOR UNRESTRICTED SEARCHES
C
      IF (IRES .EQ. Ø) THEN
        WRITE (EUNIT, 16Ø)
16Ø     FORMAT ('Ø', 'DISTRIBUTION OF SOLUTION DIP RANGES', /,
     & 1X, ' RANGE          #')
        DO 18Ø I = 1, NDELF
          WRITE (EUNIT, 17Ø) FLOAT(I - 1)*DDELF, NDRNG(I)
17Ø       FORMAT (' ', 1X, F5.1, 5X, I5)
18Ø     CONTINUE
        WRITE (EUNIT, 19Ø)
19Ø     FORMAT ('Ø', 'DISTRIBUTION OF SOLUTION STRIKE RANGES', /,
     & 1X, ' RANGE          #')
        DO 2ØØ I = 1, NPHIF
          WRITE (EUNIT, 17Ø) FLOAT(I - 1)*DPHIF, NSRNG(I)
2ØØ     CONTINUE
        WRITE (EUNIT, 21Ø)
21Ø     FORMAT ('Ø', 'DISTRIBUTION OF SOLUTION RAKE RANGES', /,
     & 1X, ' RANGE          #')
        DO 22Ø I = 1, NLAMF
          WRITE (EUNIT, 17Ø) FLOAT(I - I)*DLAMF, NRRNG(I)
22Ø     CONTINUE
      END IF
C
      WRITE (EUNIT, *) ' '
      WRITE (EUNIT, *) IEVR, ' EVENTS READ, ', IEVP, ' PROCESSED'
C
      RETURN
      END
C
C
C
C
      SUBROUTINE CSORT(CX, IX, N)
C
C  INDIRECT SORT ROUTINE FROM MEISSNER & ORGANICK, P.352
C  STORES ASCENDING SORT ORDER OF CX IN ARRAY IX, LEAVING CX UNCHANGED
C
      INTEGER           IX(*)                          ! (OUTPUT) POINTER ARRAY TO SORTED ORDER
      CHARACTER*(*)     CX(*)                          ! (INPUT) ARRAY TO BE SORT
      INTEGER           N                              ! (INPUT) NUMBER OF ELEMENTS TO BE SORTED
C
      INTEGER           I                              ! LOOP INDEX
      INTEGER           J                              ! LOOP INDEX
      INTEGER           NEXT                           ! INDEX INTO CX
C
      DO 1Ø I = 1, N
        IX(I) = I
1Ø    CONTINUE
C
      DO 4Ø J = 1, N - 1
```

```
         NEXT = IX(J + 1)
         DO 2Ø I = J, 1, -1
           IF (CX(NEXT) .GT. CX(IX(I))) GOTO 3Ø
           IX(I + 1) = IX(I)
2Ø       CONTINUE
3Ø     IX(I + 1) = NEXT
4Ø     CONTINUE
C
       RETURN
       END
C
C
C
C
C
       SUBROUTINE SEARCH (BOT, COEF, DDEL, DEL, DELC, DELØ, DFIT, DLAM,
      & DPHI, FIRST, FIT, FITMIN, FLAG, GFIT, IGOOD, IPRNT, J1, M1,
      & MXDIP, MXRAKE, MXSTAT, MXSTRK, N1, NDEL, NG, NLAM, NPHI, NR,
      & NSTAR, PHIS, PHISC, PHISØ, POBS, RAD, WTOBS, XLAM, XLAMC, XLAMØ)
C
C LOOP OVER THE ENTIRE FOCAL MECHANISM SPACE, COMPUTE FIT PARAMETER FOR EACH SOLUTION, AND RETURN BEST FIT INDICES
C IN CASE OF TIE FIT, CHOOSE FIT WITH LARGEST "BOT".  IF FINE SEARCH (FIRST = FALSE) THEN FILL IN FLAG ARRAY WITH
C STARS FOR SOLUTIONS WITH FIT PARAMETER <= BEST FIT + DFIT
C
       REAL              BOT(MXDIP,MXSTRK,MXRAKE)        ! (OUTPUT) SUM OF PRODUCT OF OBSERVED AND PREDICTED WEIGHTS
       REAL              COEF(MXSTAT,6)                  ! (INPUT) COEFFICIENTS BY WHICH TM MULTIPLIED TO GIVE P RADIATION PATTERN
       REAL              DDEL                            ! (INPUT) FAULT DIP INCREMENT IN DEGREES
       REAL              DEL(MXDIP)                      ! (OUTPUT) FAULT DIP ANGLE IN DEGREES
       REAL              DELC(MXDIP)                     ! (OUTPUT) FAULT DIP ANGLE FOR COARSE SEARCH
       REAL              DELØ                            ! (INPUT) INITIAL FAULT DIP ANGLE IN DEGREES
       REAL              DFIT                            ! (INPUT) INCREMENT TO FIT FUNCTION
       REAL              DLAM                            ! (INPUT) FAULT RAKE INCREMENT IN DEGREES
       REAL              DPHI                            ! (INPUT) FAULT STRIKE INCREMENT IN DEGREES
       LOGICAL           FIRST                           ! (INPUT) FLAG: TRUE=FIRST TIME INTO SUBROUTINE SEARCH
       REAL              FIT(MXDIP,MXSTRK,MXRAKE)        ! (OUTPUT) WEIGHTED MEASURE OF AGREEMENT BETWEEN OBS, PRED POLARITIES
       REAL              FITMIN                          ! (OUTPUT) FIT OF BEST SOLUTION CORRESPONDING TO FIT(J1, N1, M1)
       CHARACTER*1       FLAG(MXDIP,MXSTRK,MXRAKE)       ! (OUTPUT) IF FIT < FITLIM THEN '*', OTHERWISE BLANK
       REAL              GFIT(MXDIP*MXSTRK*MXRAKE)       ! (OUTPUT) FITS OF GOOD SOLUTIONS FROM COARSE SEARCH
       INTEGER           IGOOD(MXDIP*MXSTRK*MXRAKE,4)    ! (OUTPUT) ARRAY CONTAINING INDICES OF GOOD SOLUTIONS (COARSE)
       INTEGER           IPRNT                           ! (OUTPUT) FLAG: 1(Ø)=DO (NOT) PRINT OUT FIT PARAMETERS
       INTEGER           J1                              ! (OUTPUT) DIP INDEX OF BEST SOLUTION
       INTEGER           M1                              ! (OUTPUT) RAKE INDEX OF BEST SOLUTION
       INTEGER           MXDIP                           ! (INPUT) MAXIMUM # OF DIP INCREMENTS PERMITTED
       INTEGER           MXRAKE                          ! (INPUT) MAXIMUM # OF RAKE INCREMENTS PERMITTED
       INTEGER           MXSTAT                          ! (INPUT) MAXIMUM # OF STATIONS PERMITTED
       INTEGER           MXSTRK                          ! (INPUT) MAXIMUM # OF STRIKE INCREMENTS PERMITTED
       INTEGER           N1                              ! (OUTPUT) STRIKE INDEX OF BEST SOLUTION
       INTEGER           NDEL                            ! (INPUT) NUMBER OF FAULT DIP INCREMENTS
       INTEGER           NG                              ! (OUTPUT) NUMBER OF GOOD SOLUTIONS IN COARSE SEARCH
       INTEGER           NLAM                            ! (INPUT) NUMBER OF FAULT RAKE INCREMENTS
       INTEGER           NPHI                            ! (INPUT) NUMBER OF FAULT STRIKE INCREMENTS
       INTEGER           NR                              ! (INPUT) -1=EOF, Ø=NR<MINOBS, NR>Ø => NUMBER OF STATIONS
       INTEGER           NSTAR                           ! (OUTPUT) NUMBER OF SOLUTIONS HAVING FIT WITHIN 5% OF FITMIN
       REAL              PHIS(MXSTRK)                    ! (OUTPUT) FAULT STRIKE ANGLE IN DEGREES
       REAL              PHISC(MXSTRK)                   ! (OUTPUT) FAULT STRIKE ANGLE IN DEGREES FOR COARSE SEARCH
       REAL              PHISØ                           ! (INPUT) INITIAL FAULT STRIKE ANGLE IN DEGREES
       REAL              POBS(MXSTAT)                    ! (INPUT) OBSERVED FIRST MOTION POLARITIES; .5=COMPRESSION, -.5=DILATATION
       REAL              RAD                             ! (INPUT) CONVERSION FROM DEGREES TO RADIANS
       REAL              WTOBS(MXSTAT)                   ! (INPUT) OBSERVED FIRST MOTIONS WEIGHTS
       REAL              XLAM(MXRAKE)                    ! (OUTPUT) FAULT RAKE ANGLE IN DEGREES
       REAL              XLAMC(MXRAKE)                   ! (OUTPUT) FAULT RAKE ANGLE IN DEGREES FOR COARSE SEARCH
       REAL              XLAMØ                           ! (INPUT) INITIAL FAULT RAKE ANGLE IN DEGREES
C
       REAL              BEST                            ! LARGEST BOT FOR SOLUTIONS WITH FIT=FITMIN (IE. TIES)
```

```
      REAL              DIP                          ! FAULT DIP ANGLE IN RADIANS
      LOGICAL           FIRST1                       ! TEST FOR FIRST TIME INTO ROUTINE
      REAL              FITLIM                       ! UPPER BOUND ON "GOOD" SOLUTIONS IN SEARCH
      INTEGER           J                            ! LOOP INDEX OVER DIP
      INTEGER           M                            ! LOOP INDEX OVER RAKE
      INTEGER           N                            ! LOOP INDEX OF STRIKE
      REAL              PRAD                         ! RADIATION AMPLITUDE CORRESPONDING AIN, PHI.
                                                     ! (DILATATION) -1.<PRAD<+1.(COMPRESSION)
      REAL              PTH                          ! PREDICTED FIRST MOTION POLARITY; SAME CONVENTION AS FOR POBS
      REAL              SLIP                         ! FAULT SLIP ANGLE IN RADIANS
      REAL              STRIKE                       ! FAULT STRIKE ANGLE IN RADIANS
      REAL              TM(6)                        ! MOMENT TENSOR IN UPPER TRIANGULAR SYMETRIC STORAGE MODE
      REAL              TOP                          ! SUM OF WEIGHTED DIFFERENCE OF PREDICTED, OBS. POLARITIES; Ø<= TOP <=1
      REAL              WTTH                         ! PREDICTED FIRST MOTIONS WEIGHTS
C
      BEST = Ø.
      FITMIN = 2.Ø
      DO 5Ø M = 1, NLAM
        XLAM(M) = XLAMØ + (M - 1)*DLAM
        IF (FIRST) XLAMC(M)=XLAM(M)
        DO 4Ø N = 1, NPHI
          PHIS(N) = PHISØ + (N - 1) * DPHI
          IF (FIRST) PHISC(N)=PHIS(N)
          DO 3Ø J = 1, NDEL
            DEL(J) = DELØ + (J - 1) * DDEL
            IF (FIRST) DELC(J)=DEL(J)
            STRIKE = PHIS(N)*RAD
            DIP = DEL(J)*RAD
            SLIP = XLAM(M)*RAD
C
C CALCULATE MOMENT TENSOR REPRESENTATION OF SHEAR FAULT (POSTIVE UP, SOUTH, EAST)
C
            CALL SHRFLT (STRIKE, DIP, SLIP, TM)
C
C CALCULATE RADIATION PATTERN FOR MODEL (EQTN 4.91, AKI & RICHARDS, PG. 118)
C
            TOP = Ø
            BOT(J, N, M) = Ø
            DO 2Ø I = 1, NR
              PRAD = Ø
              DO 1Ø K = 1, 6
                PRAD = PRAD + TM(K)*COEF(I, K)
1Ø            CONTINUE
C
C CALCULATE FIT FUNCTION FOR THIS MODEL
C
              PTH = SIGN(Ø.5, PRAD)
              WTTH = SQRT(ABS(PRAD))
              TOP = TOP + ABS(POBS(I) - PTH)*WTOBS(I)*WTTH
              BOT(J, N, M) = BOT(J, N, M) + WTOBS(I)*WTTH
2Ø          CONTINUE
            FIT(J, N, M) = TOP/BOT(J, N, M)
            IF (FIT(J, N, M) .LT. FITMIN) FITMIN = FIT(J, N, M)
3Ø        CONTINUE
4Ø      CONTINUE
5Ø    CONTINUE
C
C FOR TIE SOLUTIONS, FIND SOLUTION WITH MOST STATIONS AWAY FROM NODES
C
      FITLIM = FITMIN + DFIT
      NG = Ø
      NSTAR = -1
      DO 9Ø M = 1, NLAM
        DO 8Ø N = 1, NPHI
```

```
          DO 7Ø J = 1, NDEL
            IF (FIT(J, N, M) .EQ. FITMIN .AND. BOT(J, N, M) .GT. BEST)
     & THEN
              BEST = BOT(J, N, M)
              J1 = J
              N1 = N
              M1 = M
            END IF
C
C SAVE SOLUTIONS IN COARSE SEARCH WITH FIT .LE. FITLIM AS "GOOD" SOLUTIONS
C
            IF (FIRST) THEN
              IF (FIT(J, N, M) .LE. FITLIM) THEN
                NG = NG + 1
                IGOOD(NG, 1) = J
                IGOOD(NG, 2) = N
                IGOOD(NG, 3) = M
                IGOOD(NG, 4) = Ø
                GFIT(NG) = FIT(J, N, M)
              END IF
C
C STAR SOLUTIONS HAVING FIT WITHIN DFIT OF FITMIN ON FINE SEARCH
C
            ELSE
              IF (FIT(J, N, M) .LE. FITLIM) THEN
                FLAG(J, N, M) = '*'
                NSTAR = NSTAR + 1
              ELSE
                FLAG(J, N, M) = ' '
              END IF
            END IF
7Ø        CONTINUE
8Ø      CONTINUE
9Ø    CONTINUE
      IF (.NOT. FIRST) FLAG(J1, N1, M1) = 'A'
C
      RETURN
      END
C
C
C
C
C
        SUBROUTINE PEXCF (COEF, I, MXSTAT, U)
C
C
C CALCULATES COEFFICIENTS FOR DETERMINING THE FAR-FIELD RADIATION PATTERN OF P WAVES FROM THE MOMENT-RATE TENSOR COMPONENTS OF A
C POINT SOURCE IN AN INFINITE, HOMOGENEOUS, ELASTIC MEDIUM.  THE RADIATION PATTERN IS NORMALIZED; TO OBTAIN PARTICLE AMPLITUDES,
C MULTIPLY BY
C
C    1.Ø/(4.Ø*PI*RHO*(V**3)*R),
C
C     WHERE:
C          RHO IS THE DENSITY IN THE SOURCE REGION,
C          V IS THE P-WAVE SPEED IN THE SOURCE REGION, AND
C          R IS THE GEOMETRIC SPREADING FACTOR
C            (FOR A HOMOGENEOUS MEDIUM, THIS IS EQUAL TO THE DISTANCE
C            TO THE OBSERVATION POINT.)
C
C REFERENCE:
C         AKI, KEIITI, AND PAUL G. RICHARDS, QUANTITATIVE SEISMOLOGY,
C         FREEMAN, SAN FRANCISCO, 198Ø, EQUATION 49.1, PAGE 118.
C
C WRITTEN BY BRUCE JULIAN
```

```
C
      REAL              COEF(MXSTAT, 6)                ! (OUTPUT) EXCITATION COEFFICIENTS
      INTEGER           I                              ! (INPUT) INDEX OF STATION
      INTEGER           MXSTAT                         ! (INPUT) MAXIMUM # OF STATIONS PERMITTED
      REAL              U(3)                           ! (INPUT) UNIT VECTOR IN RAY DIRECTION
C
      COEF(I, 1) = U(1)*U(1)
      COEF(I, 2) = 2.*U(1)*U(2)
      COEF(I, 3) = U(2)*U(2)
      COEF(I, 4) = 2.*U(3)*U(1)
      COEF(I, 5) = 2.*U(2)*U(3)
      COEF(I, 6) = U(3)*U(3)
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE SHRFLT(STRIKE, DIP, SLIP, TM)
C
C    THIS SUBROUTINE CALCULATES THE MOMENT-TENSOR REPRESENTATION OF A SHEAR FAULT, GIVEN ITS STRIKE, DIP, AND SLIP ANGLES.
C
C  METHOD:
C    THE MOMENT TENSOR IS FIRST EXPRESSED IN A COORDINATE SYSTEM WITH THE Z AXIS NORMAL TO THE FAULT PLANE AND THE X AXIS IN
C     THE SLIP DIRECTION:
C
C                  (Ø.  Ø.  1.)
C                  (Ø.  Ø.  Ø.)
C                  (1.  Ø.  Ø.)
C
C    THIS COORDINATE SYSTEM IS THEN ROTATED THROUGH THE EULER ANGLES PHI = -SLIP, THETA = -DIP, AND PSI = STRIKE - PI,
C    (CONVENTIONS OF GOLDSTEIN, CLASSICAL MECHANICS, SEC 4-4) WHICH RESULTS IN A (SOUTH, EAST, UP) ORIENTATION OF THE (X, Y, Z)
C    AXES, RESPECTIVELY.  A PERMUTATION THEN CONVERTS THIS TO THE ORDER (UP, SOUTH, EAST).  THE STRENGTH OF THE DOUBLE-COUPLE IS
C    TAKEN AS UNITY; THE CALCULATED MOMENT TENSOR COMPONENTS MUST BE MULTIPLIED BY THE FACTOR:
C
C        MU*A*S
C
C    WHERE:
C        MU IS THE RIGIDITY MODULUS OF THE MEDIUM
C        A IS THE FAULT AREA, AND
C        S IS THE MEAN DISLOCATION ACROSS THE FAULT.
C          (NOTE:  IF THE MEAN DISLOCATION VELOCITY IS USED INSTEAD,
C          THE RESULT WILL BE THE MOMENT-RATE TENSOR.)
C
C    WRITTEN BY BRUCE R. JULIAN ON 7 APRIL, 1977.
C
      REAL              DIP                            ! (INPUT) FAULT DIP ANGLE IN RADIANS
      REAL              SLIP                           ! (INPUT) FAULT SLIP ANGLE IN RADIANS
      REAL              STRIKE                         ! (INPUT) FAULT STRIKE ANGLE IN RADIANS
      REAL              TM(6)                          ! (OUTPUT) SEISMIC MOMENT TENSOR ARRANGED IN THE FOLLOWING ORDER:
                                                       ! (R, R)           I.E. (UP, UP)
                                                       ! (R, THETA)       I.E. (UP, SOUTH)
                                                       ! (THETA, THETA)   I.E. (SOUTH, SOUTH)
                                                       ! (R, PHI)         I.E. (UP, EAST)
                                                       ! (THETA, PHI)     I.E. (SOUTH, EAST)
                                                       ! (PHI, PHI)       I.E. (EAST, EAST)
C
      REAL              A11                            !  TRANSFORMATION MATRIX
      REAL              A21                            !  TRANSFORMATION MATRIX
      REAL              A31                            !  TRANSFORMATION MATRIX
      REAL              A13                            !  TRANSFORMATION MATRIX
      REAL              A23                            !  TRANSFORMATION MATRIX
```

```
      REAL             A33                              ! TRANSFORMATION MATRIX
      REAL             CD                               ! COS(DIP)
      REAL             CL                               ! COS(SLIP)
      REAL             CS                               ! COS(STRIKE)
      REAL             SD                               ! SIN(DIP)
      REAL             SL                               ! SIN(SLIP)
      REAL             SS                               ! SIN(STRIKE)
C
C  CALCULATE COMPONENTS OF ORTHOGONAL TRANSFORMATION MATRIX
C  FROM FAULT-ORIENTED TO (SOUTH, EAST, UP) COORDINATE SYSTEM
C
      SS = SIN(STRIKE)
      CS = COS(STRIKE)
      SD = SIN(DIP)
      CD = COS(DIP)
      SL = SIN(SLIP)
      CL = COS(SLIP)
      A11 = -CS*CL - CD*SL*SS
      A21 =  SS*CL - CD*SL*CS
      A31 = SD*SL
      A13 = SS*SD
      A23 = CS*SD
      A33 = CD
C
C  TRANSFORM MOMENT TENSOR (Ø,   Ø,   1,
C                           Ø,   Ø,   Ø,
C                           1,   Ø,   Ø)
C
C  AND PERMUTE AXES TO (UP, SOUTH, EAST) ORDER
C
      TM(1) = 2*A31*A33
      TM(2) = A11*A33 + A31*A13
      TM(3) = 2*A11*A13
      TM(4) = A21*A33 + A31*A23
      TM(5) = A11*A23 + A21*A13
      TM(6) = 2*A21*A23
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE REFRMT (DEL, IDIP, IDIPDR, ISLIP, PHIS, XLAM)
C
C  REFORMAT DIP, STRIKE, AND RAKE ANGLES TO INTEGER VALUES AND CONVERT STRIKE TO DOWN-DIP DIRECTION
C
      REAL             DEL                              ! (INPUT) FAULT DIP ANGLE IN DEGREES
      INTEGER          IDIP                             ! (OUTPUT) FAULT DIP ANGLE IN DEGREES
      INTEGER          IDIPDR                           ! (OUTPUT) DIP DIRECTION IN DEGREES
      INTEGER          ISLIP                            ! (OUTPUT) RAKE IN DEGREES
      REAL             PHIS                             ! (INPUT) FAULT STRIKE ANGLE IN DEGREES
      REAL             XLAM                             ! (INPUT) FAULT RAKE ANGLE IN DEGREES
C
      INTEGER          ISTRK                            ! STRIKE OF BEST FIT
C
      IDIP = IFIX(DEL)
      ISTRK = IFIX(PHIS)
      ISLIP = IFIX(XLAM)
      IF (IDIP .GT. 9Ø) THEN
        IDIP = 18Ø - IDIP
        ISTRK = ISTRK + 18Ø
        ISLIP = -ISLIP
```

```
      ELSE IF (IDIP .LT. Ø) THEN
        IDIP = -IDIP
        ISTRK = ISTRK + 18Ø
        ISLIP = ISLIP + 18Ø
      END IF
      IDIPDR = JMOD(ISTRK + 9Ø, 36Ø)
      IF (IDIPDR .LT. Ø) IDIPDR = IDIPDR + 36Ø
      ISLIP = JMOD(ISLIP, 36Ø)
      IF (ISLIP .GT. 18Ø) ISLIP = ISLIP - 36Ø
      IF (ISLIP .LT. -18Ø) ISLIP = ISLIP + 36Ø
C
      RETURN
      END
C
C
C
C
C
      LOGICAL FUNCTION COMPL (SOLNS, NSOL, DD, DA, SA, AERR, MXSLNS)
C
C THIS FUNCTION COMPARES A "NEW" FAULT PLANE SOLUTION (DD, DA, SA) WITH A LIST OF OTHER FAULT PLANE SOLUTIONS
C  AND CHECKS FOR ANY OF THE FOLLOWING CONDITIONS
C
C   1. THE "NEW" SOLUTION IS SIMILAR TO ONE OF THE SOLUTIONS IN SOLNS
C   2. THE COMPLIMENT OF THE "NEW" SOLUTION IS SIMILAR TO ONE OF THE SOLUTIONS IN SOLNS
C   3. THE "NEW" SOLUTION IS SIMILAR TO THE COMPLIMENT OF ONE OF THE SOLUTIONS IN SOLNS
C
C IF ANY ONE OF THE ABOVE CONDITIONS IS TRUE, FUNCTION COMPL RETURNS WITH A VALUE .TRUE.
C OTHERWISE, FUNCTION COMPL RETURNS WITH THE VALUE .FALSE.
C
C  SOLUTIONS ARE SIMILAR IF ALL THREE PAIRS OF CORRESPONDING ANGLES DIFFER BY LESS THAN AERR.
C
      REAL              AERR                      ! ALLOWABLE DIFFERENCE BETWEEN CORRESPONDING ANGLES OF COMPLIMENTARY PLANES
      REAL              DA                        ! DIP ANGLE OF NEW PLANE
      REAL              DD                        ! DIP DIRECTION ANGLE OF NEW PLANE
      INTEGER           MXSLNS                    ! MAXIMUM # OF MULTIPLE SOLUTIONS PERMITTED
      INTEGER           NSOL                      ! NUMBER OF PLANES STORED IN ARRAY SOLNS
      REAL              SA                        ! SLIP ANGLE OF NEW PLANE
      REAL              SOLNS(MXSLNS,3)           ! ARRAY OF PLANES (DD, DA, SA) TO TEST NEW PLANE AGAINST
C
      INTEGER           J                         ! LOOP INDEX
      REAL              AUX1(3)                   ! DIP DIRECTION, ANGLE, AND RAKE OF AUXILLIARY PLANE OF NEW PLANE
      REAL              AUX2(3)                   ! DIP DIRECTION, ANGLE, AND RAKE OF AUXILLIARY PLANE OF PRREVIOUS PLANES
C
      COMPL = .FALSE.
C
      CALL AUXPLN (DD, DA, SA, AUX1(1), AUX1(2), AUX1(3))
C
      DO 4Ø J = 1, NSOL
C
C COMPARE NEW SOLUTION WITH EACH SOLUTION ON LIST
C
        IF (ABS(DD - SOLNS(J, 1)) .LT. AERR .AND.
     &ABS(DA - SOLNS(J, 2)) .LT. AERR .AND.
     &RDIFF(SA, SOLNS(J, 3)) .LT. AERR)  THEN
          COMPL = .TRUE.
          RETURN
        END IF
C
C      COMPARE COMPLIMENT OF "NEW SOLUTION" WITH EACH SOLUTION ON LIST
C
        IF (ABS(SOLNS(J, 1) - AUX1(1)) .LT. AERR .AND.
     &ABS(SOLNS(J, 2) - AUX1(2)) .LT. AERR .AND.
     &RDIFF(SOLNS(J, 3), AUX1(3)) .LT. AERR) THEN
```

```
          COMPL = .TRUE.
          RETURN
        END IF
C
        CALL AUXPLN (SOLNS(J, 1), SOLNS(J, 2), SOLNS(J, 3), AUX2(1),
     & AUX2(2), AUX2(3))
C
C      COMPARE "NEW SOLUTION" WITH COMPLIMENT OF EACH SOLUTION ON LIST
C
        IF (ABS(DD - AUX2(1)) .LT. AERR .AND.
     &ABS(DA - AUX2(2)) .LT. AERR .AND.
     &RDIFF(SA, AUX2(3)) .LT. AERR) THEN
          COMPL = .TRUE.
          RETURN
        END IF
40    CONTINUE
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE AUXPLN (DD1, DA1, SA1, DD2, DA2, SA2)
C
C     CALCULATE THE AUXILLIARY PLANE OF A DOUBLE COUPLE FAULT PLANE SOLUTION, GIVEN THE PRINCIPLE PLANE.
C
C     WRITTEN BY PAUL REASENBERG, JUNE, 1984, FROM CLASS NOTES BY DAVE BOORE, (BOTH AT THE U.S.G.S., MENLO PARK.)
C     ANGLE VARIABLES PHI, DEL AND LAM ARE AS DEFINED IN AKI AND RICHARDS, (1980), P.114.
C
      REAL              DA1                               ! (INPUT)  DIP ANGLE IN DEGREES OF PRICIPLE PLANE
      REAL              DD1                               ! (INPUT)  DIP DIRECTIONS IN DEGREES OF PRICIPLE PLANE
      REAL              SA1                               ! (INPUT)  SLIP ANGLE IN DEGREES OF PRICIPLE PLANE
      REAL              DA2                               ! (OUTPUT)  DIP ANGLE IN DEGREES OF AUXILLIARY PLANE
      REAL              DD2                               ! (OUTPUT)  DIP DIRECTIONS IN DEGREES OF AUXILLIARY PLANE
      REAL              SA2                               ! (OUTPUT)  SLIP ANGLE IN DEGREES OF AUXILLIARY PLANE
C
      DOUBLE PRECISION  BOT                               ! SCRATCH VARIABLE
      DOUBLE PRECISION  DEL1                              ! DIP ANGLE OF PRINCIPAL PLANE IN RADIANS
      LOGICAL           FIRST                             ! TEST: TRUE IF FIRST TIME INTO ROUTINE
      DOUBLE PRECISION  PHI1                              ! FAULT PLANE STRIKE OF PRINCIPAL PLANE
      DOUBLE PRECISION  PHI2                              ! STRIKE OF AUXILLIARY PLANE IN RADIANS
      DOUBLE PRECISION  RAD                               ! CONVERSION FACTOR FROM DEGREES TO RADIAN
      DOUBLE PRECISION  SGN                               ! SAVES PRINCIPAL PLANE SLIP ANGLE FOR ASSIGNING PROPER SIGN TO AUXILLIARY
      DOUBLE PRECISION  TOP                               ! SCRATCH VARIABLE
      DOUBLE PRECISION  XLAM1                             ! SLIP ANGLE OF PRINCIPAL PLANE IN RADIANS
      DOUBLE PRECISION  XLAM2                             ! SLIP ANGLE OF AUXILLIARY PLANE
C
      DATA FIRST /.TRUE./
      SAVE FIRST, RAD
C
      IF (FIRST) THEN
        FIRST = .FALSE.
        RAD = DATAN(1.0D0)/45.0D0
      END IF
C
      PHI1 = DD1 - 90.0D0
      IF (PHI1 .LT. 0.0D0) PHI1 = PHI1 + 360.0D0
      PHI1 = PHI1*RAD
      DEL1 = DA1*RAD
      SGN = SA1
      XLAM1 = SA1*RAD
```

```
C
      TOP = DCOS(XLAM1)*DSIN(PHI1) - DCOS(DEL1)*DSIN(XLAM1)*DCOS(PHI1)
      BOT = DCOS(XLAM1)*DCOS(PHI1) + DCOS(DEL1)*DSIN(XLAM1)*DSIN(PHI1)
      DD2 = DATAN2(TOP, BOT)/RAD
      PHI2 = (DD2 - 9Ø.ØDØ)*RAD
      IF (SA1 .LT. Ø.ØDØ) DD2 = DD2 - 18Ø.ØDØ
      IF (DD2 .LT. Ø.ØDØ) DD2 = DD2 + 36Ø.ØDØ
      IF (DD2. GT. 36Ø.ØDØ) DD2 = DD2 - 36Ø.ØDØ
C
      DA2 = DACOS(DSIN(DABS(XLAM1))*DSIN(DEL1))/RAD
      XLAM2 = -DCOS(PHI2)*DSIN(DEL1)*DSIN(PHI1) +
     & DSIN(PHI2)*DSIN(DEL1)*DCOS(PHI1)
C
C MACHINE ACCURACY PROBLEM
C
      IF (ABS(XLAM2) .GT. 1.ØDØ) THEN
        XLAM2 = DSIGN(1.ØDØ, XLAM2)
      END IF
      XLAM2 = DSIGN(DACOS(XLAM2), SGN)
      SA2 = XLAM2/RAD
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE HHOG (EUNIT, JSTRT, NSTRT, MSTRT, IGOOD, GFIT, NG,
     & IDST, NDST, MXDIP, MXSLNS, MXSTRK, MXRAKE, NDELC, NPHIC, NLAMC)
C
C  PERFORMS A "HEDGEHOG" SEARCH THROUGH COARSE SOLUTIONS WITH FITS LESS THAN FITLIM, IDENTIFIES SOLUTIONS BELONGING TO
C  DISCRETE LOCALIZED MINIMA, AND RETURNS STRIKE, DIP, AND RAKE INDICES OF SOLUTION WITH BEST FIT WITHIN EACH MINIMA
C
      INTEGER           MXHOG                          ! MAXIMUM NUMBER OF SOLUTIONS PER LOCALIZED MINIMA
C
      PARAMETER (MXHOG = 2ØØ)
C
      INTEGER           EUNIT                          ! (INPUT) LOGICAL UNIT # OF OUTPUT OF ERROR MESSAGES
      REAL              GFIT(MXDIP*MXSTRK*MXRAKE)      ! (INPUT) CONTAINS FITS OF SOLUTIONS IN IGOOD
      INTEGER           IDST(MXSLNS,3)                 ! (OUTPUT INDICES OF BEST FITTING SOLUTIONS IN EACH LOCALIZED MINIMA
      INTEGER           IGOOD(MXDIP*MXSTRK*MXRAKE,4)   ! (INPUT) INDICES OF SOLUTIONS WITH "GOOD" FITS DETERMINED BY COARSE SEARCH.
      INTEGER           JSTRT                          ! (INPUT) DIP INDEX OF BEST SOLUTION FROM COARSE SEARCH
      INTEGER           MSTRT                          ! (INPUT) RAKE INDEX OF BEST SOLUTION FROM COARSE SEARCH
      INTEGER           MXDIP                          ! (INPUT) MAXIMUM NUMBER OF DIP VALUES IN SEARCH
      INTEGER           MXRAKE                         ! (INPUT) MAXIMUM NUMBER OF RAKE VALUES IN SEARCH
      INTEGER           MXSLNS                         ! (INPUT) MAXIMUM NUMBER OR MULTIPLE SOLUTIONS PERMITTED
      INTEGER           MXSTRK                         ! (INPUT) MAXIMUM NUMBER OR STRIKE VALUES IN SEARCH
      INTEGER           NDELC                          ! (INPUT) NUMBER OF INCREMENTS OF DIP IN COARSE SEARCH
      INTEGER           NDST                           ! (OUTPUT NUMBER OF SOLUTIONS IN IDST
      INTEGER           NG                             ! (INPUT) NUMBER OF SOLUTIONS IN IGOOD
      INTEGER           NLAMC                          ! (INPUT) NUMBER OF INCREMENTS OF RAKE IN COARSE SEARCH
      INTEGER           NPHIC                          ! (INPUT) NUMBER OF INCREMENTS OF STRIKE IN COARSE SEARCH
      INTEGER           NSTRT                          ! (INPUT) STRIKE INDEX OF BEST SOLUTION FROM COARSE SEARCH
C
      REAL              FMIN                           ! SMALLEST FIT VALUE WITHIN SET OF SOLUTIONS COMPRISING A LOCALIZED MINIMA
      INTEGER           IC                             ! NUMBER OF SOLUTIONS USED AS CENTER POINT FOR EXPANSION
      INTEGER           ICACH(MXHOG)                   ! POINTER ARRAY INDICES OF IGOOD
      INTEGER           ICT                            ! TOTAL NUMBER OF SOLUTIONS IN A HEDGEHOG
      INTEGER           IG                             ! INDEX OVER IGOOD
      INTEGER           IK                             ! LOOP INDEX OVER ICACH
      INTEGER           JØ                             ! DIP INDEX OF CENTER POINT FOR EXPANSION
      INTEGER           JJ                             ! DIP INDEX OF NEARBY SOLUTION TO CENTERPOINT
      INTEGER           JJX                            ! DIP INDEX OF NEARBY SOLUTION TO CENTERPOINT
```

```
      INTEGER           MØ                              ! RAKE INDEX OF CENTER POINT FOR EXPANSION
      INTEGER           MM                              ! RAKE INDEX OF NEARBY SOLUTION TO CENTERPOINT
      INTEGER           MMX                             ! RAKE INDEX OF NEARBY SOLUTION TO CENTERPOINT
      INTEGER           NØ                              ! STRIKE INDEX OF CENTER POINT FOR EXPANSION
      INTEGER           NHH                             ! HEDGEHOG INDEX
      INTEGER           NN                              ! STRIKE INDEX OF NEARBY SOLUTION TO CENTERPOINT
      INTEGER           NNX                             ! STRIKE INDEX OF NEARBY SOLUTION TO CENTERPOINT
C
      NHH = 1
      IC = Ø
      ICT = Ø
      JØ = JSTRT
      NØ = NSTRT
      MØ = MSTRT
C
C EXPAND ABOUT (JØ, NØ, MØ) FOR NEAREST NEIGHBORS
C
2Ø    DO 9Ø JJ = JØ - 1, JØ + 1
        IF (JJ .LE. Ø) THEN
          JJX = NDELC - JJ
        ELSE IF (JJ .GT. NDELC) THEN
          JJX = JJ - NDELC
        ELSE
          JJX = JJ
        END IF
        DO 8Ø NN = NØ - 1, NØ + 1
          IF (NN .LE. Ø) THEN
            NNX = NPHIC - NN
          ELSE IF (NN .GT. NPHIC) THEN
            NNX = NN - NPHIC
          ELSE
            NNX = NN
          END IF
          DO 7Ø MM = MØ - 1, MØ + 1
            IF (MM .LE. Ø) THEN
              MMX = NLAMC - MM
            ELSE IF (MM .GT. NLAMC) THEN
              MMX = MM - NLAMC
            ELSE
              MMX = MM
            END IF
C
C LOOK UP EACH SOLUTION IN IGOOD. IF FOUND, ANNOTATE IT WITH THE CURRENT VALUE OF NHH
C
            DO 6Ø IG = 1, NG
              IF (IGOOD(IG, 1) .EQ. JJX .AND. IGOOD(IG, 2) .EQ. NNX
     & .AND. IGOOD(IG, 3) .EQ. MMX .AND. IGOOD(IG, 4) .EQ. Ø) THEN
              IGOOD(IG, 4) = NHH
C
C CHECK TO SEE IF THIS SOLUTION IS ALREADY IN A CACHE
C
                IF (ICT .GT. Ø) THEN
                  DO 5Ø IK = 1, ICT
                    IF (ICACH(IK) .EQ. IG) GOTO 7Ø
5Ø                CONTINUE
                END IF
C
C STORE THIS SOLUTION IN CACHE
C
                ICT = ICT + 1
                IF (ICT .GT. MXHOG) THEN
                  WRITE (EUNIT, *) '***** HHOG ERROR: NUMBER OF SOLUTION
     &S WITHIN HEDGEHOG EXCEEDS ', MXHOG, ' *****'
                  STOP
```

```
                END IF
                ICACH(ICT) = IG
              END IF
60          CONTINUE
70        CONTINUE
80      CONTINUE
90    CONTINUE
C
C SELECT NEXT SOLUTION WITHIN CURRENT HEDGEHOG AS STARTING POINT FOR EXPANSION
C
      IC = IC + 1
      IF (IC .LE. ICT) THEN
        J0 = IGOOD(ICACH(IC), 1)
        N0 = IGOOD(ICACH(IC), 2)
        M0 = IGOOD(ICACH(IC), 3)
        GOTO 20
      ELSE
C
C FINISHED PROCESSING CACHE FOR CURRENT HEDGEHOG
C
        ICT = 0
        IC = 0
        NHH = NHH + 1
        IF (NHH .GT. MXSLNS) THEN
          PRINT *, '***** HHOG ERROR: NUMBER OF MULTIPLE SOLUTIONS EXCEE
     &DS ', MXSLNS, ' *****'
          STOP
        END IF
C
C GET NEXT SOLUTION FROM IGOOD THAT DOES NOT ALREADY BELONG TO A MINIMA
C
        DO 100 IG = 1, NG
          IF (IGOOD(IG, 4) .EQ. 0) THEN
            J0 = IGOOD(IG, 1)
            N0 = IGOOD(IG, 2)
            M0 = IGOOD(IG, 3)
            GOTO 20
          END IF
100     CONTINUE
      END IF
C
C IDENTIFY SOLUTION CORRESPONDING TO FIT MINIMUM WITHIN EACH HEDGEHOG
C
      DO 130 NDST = 1, NHH - 1
        FMIN = 1.
        DO 120 IG = 1, NG
          IF (IGOOD(IG, 4) .EQ. NDST .AND. GFIT(IG) .LE. FMIN) THEN
            IDST(NDST, 1) = IGOOD(IG, 1)
            IDST(NDST, 2) = IGOOD(IG, 2)
            IDST(NDST, 3) = IGOOD(IG, 3)
            FMIN = GFIT(IG)
          END IF
120     CONTINUE
130   CONTINUE
      NDST = NHH - 1
C
      RETURN
      END
C
C
C
C
C
      FUNCTION RDIFF (RAKE1, RAKE2)
```

```
C
C  RETURNS WITH THE SMALLEST ABSOLUTE DIFFERENCE IN SLIP ANGLE BETWEEN RAKE1 AND RAKE2.
C
C  RAKE CONVENTION FOLLOWS AKI & RICHARDS, 198Ø, QUANTITATIVE SEISMOLOGY, P. 114
C
      REAL              RAKE1                                ! (INPUT) FIRST RAKE
      REAL              RAKE2                                ! (INPUT) SECOND RAKE
C
      REAL              A                                    ! STORES FIRST RAKE
      REAL              B                                    ! STORES SECOND RAKE
      REAL              C                                    ! STORES RAKE DIFFERENCE
C
      RDIFF = 999.
      A = RAKE1
      IF (RAKE1 .LT. Ø.) A = 36Ø. + RAKE1
      B = RAKE2
      IF (RAKE2 .LT. Ø.) B = 36Ø. + RAKE2
      C = ABS(A - B)
      IF (C .GT. 18Ø.) C = 36Ø. - C
      RDIFF = C
      RETURN
      END
```

```
      PROGRAM FPPLOT
C
C
C     VERSION 1.0  -  OCTOBER 31, 1985
C
C     PURPOSE:        PLOT EARTHQUAKE RAY POLARITIES AND FAULT PLANES ON A LOWER HEMISPHERE EQUAL AREA PROJECTION.
C
C     INPUT FILE:     A FILE OF THE TYPE "*.POL", WHICH IS GENERATED BY THE PROGRAM "FPFIT" (SEE "FPFIT, FPPLOT, AND FPPAGE: FORTRAN
C                     COMPUTER PROGRAMS FOR CALCULATING AND DISPLAYING EARTHQUAKE FAULT-PLANE SOLUTIONS,
C                     BY P. REASENBERG AND D. OPPENHEIMER, U.S. GEOLOGICAL SURVEY OPEN-FILE REPORT 85-???)
C
C     REQUIRED ROUTINES: CALCOMP STYLE PLOT ROUTINES PLOTS, PLOT, NEWPEN, SYMBOL.
C
C     DEPARTURES FROM FORTRAN-77 STANDARD:
C                     1. KEYWORD "READONLY" OPEN STATEMENT
C                     2. EMBEDDED COMMENTS PREFACED WITH AN EXCLAMATION MARK (!) FOLLOWING VARIABLE DECLARATIONS
C                     3. ARGUMENT LIST BUILT-IN FUNCTION "%REF"
C                     4. CALLS TO VAX SYSTEM ROUTINES "SYS$BINTIM", "SYS$SCHDWK", "SYS$HIBER"  (SUBROUTINE DELAY)
C
C      OUTPUT:        GRAPHIC OUTPUT ONLY
C
C      AUTHORS:       PAUL REASENBERG AND DAVID OPPENHEIMER, U.S.G.S. IN MENLO PARK.  SOME OF THE ROUTINES
C                     WERE ADOPTED FROM CODE WRITTEN BY JOHN LAHR, BRUCE JULIAN, AND FRED KLEIN.
C
      REAL            AIN                               ! RAY ANGLE OF INCIDENCE
      CHARACTER*1     ANS                               ! FLAG: Y(N)=DO (NOT) PLOT STATION NAMES ADJACENT TO FIRST MOTIONS
      REAL            AZM                               ! RAY AZIMUTH
      REAL            CX1                               ! X POSITION OF LARGE CIRCLE CENTER
      REAL            CX2                               ! X POSITION OF SMALL CIRCLE CENTER
      REAL            CY1                               ! Y POSITION OF LARGE CIRCLE CENTER
      REAL            CY2                               ! Y POSITION OF SMALL CIRCLE CENTER
      REAL            DA                                ! DIP ANGLE
      REAL            DAN(11)                           ! DIP ANGLE OF NEARBY SOLUTIONS
      REAL            DD                                ! DIP DIRECTION
      REAL            DDN(11)                           ! DIP DIRECTION OF NEARBY SOLUTIONS
      CHARACTER*1     DISC                              ! FLAG: IF "*" THEN FIRST MOTION DISCREPANT WITH SOLUTION
      REAL            DIST                              ! EPICENTRAL DISTANCE
      CHARACTER*132   EVENT                             ! HYPO71 SUMMARY CARD
      CHARACTER*50    FILNAM                            ! FILE NAME OF DATA
      LOGICAL         FIRST                             ! FLAG: TRUE PRIOR TO PLOTTING DISCREPANT FIRST MOTION
      REAL            HITE1                             ! HEIGHT OF P,T SYMBOL IN LARGE CIRCLE
      REAL            HITE2                             ! HEIGHT OF P,T SYMBOL IN SMALL CIRCLE, DSCRPNCY RPRT
      REAL            HITE3                             ! HEIGHT OF TITLE, EXTENDED SUMMARY CARD, FIRST MOTION LEGEND
      INTEGER         I                                 ! LOOP INDEX OVER NUMBER OF LINES OF NEARBY SOLUTIONS
      INTEGER         INDX                              ! CHARACTER POSITION OF FIRST BLANK IN TITLE
      INTEGER         IOS                               ! I/O STATUS DESCRIPTOR
      CHARACTER*40    LINE                              ! SCRATCH VARIABLE FOR PLOT OUTPUT
      CHARACTER*4     NAME                              ! STATION NAME
      INTEGER         N                                 ! LOOP INDEX OVER NUMBER OF NEARBY SOLUTIONS
      INTEGER         NEV                               ! EVENT NUMBER
      INTEGER         NLINE                             ! NUMBER OF LINES OF NEARBY SOLUTIONS
      INTEGER         NREAD                             ! NUMBER OF NEARBY SOLUTIONS PER LINE
      INTEGER         NSKIP                             ! NUMBER OF SOLUTIONS TO SKIP
      INTEGER         NSTAR                             ! NUMBER OF SOLUTIONS WITH FIT WITHIN  5% OF BEST SOLUTION
      REAL            PI                                ! PI
      CHARACTER*4     PRMK                              ! FIRST MOTION DESCRIPTION (EG. IPU0)
      REAL            RAD                               ! PI/180
      REAL            RMAX1                             ! RADIUS OF LARGE CIRCLE
      REAL            RMAX2                             ! RADIUS OF SMALL CIRCLE
      REAL            SA                                ! RAKE
      REAL            SAN(11)                           ! RAKE OF NEARBY SOLUTIONS
      CHARACTER*1     SYM                               ! FIRST MOTION DIRECTION
      CHARACTER*80    TITLE                             ! DATA SET DESCRIPTOR
      REAL            WT                                ! WEIGHT ASSIGNED TO PICK QUALITY IN PROGRAM FPFIT
      REAL            XPOS1                             ! LEFTMOST X POSITION OF TITLE, SUMMARY CARD, SYMBOL LEGEND
      REAL            XPOS2                             ! LEFTMOST X POSITION OF DISCREPANCY REPORT
      REAL            YPOS                              ! Y PLOT POSITION
      REAL            YPOS1                             ! Y PLOT POSITION OF TITLE
      REAL            YPOS2                             ! Y PLOT POSITION OF SYMBOL LEGEND
      REAL            YPOS3                             ! Y PLOT POSITION OF TOP OF DISCREPANCY REPORT
```

```
      PARAMETER (CX1 = 3.Ø, CX2 = 7.25, CY1 = 3.Ø, CY2 = 1.25)
      PARAMETER (HITE1 = Ø.2, HITE2 = Ø.Ø7, HITE3 = Ø.1, RMAX1 = 2.95)
      PARAMETER (RMAX2 = 1.25, XPOS1 = Ø.1, XPOS2 = 6.5)
      PARAMETER (YPOS1 = 7.3, YPOS2 = Ø.1, YPOS3 = 6.Ø)
      PI = ATAN(1.Ø)*4.Ø
      RAD = PI/18Ø.Ø
C
1Ø    WRITE (6, 15) 'ENTER NAME OF DATA FILE:  '
15    FORMAT ('$', A)
      READ (5, 16, ERR = 1Ø) FILNAM
16    FORMAT (A)
      OPEN (UNIT = 2, FILE = FILNAM, ERR = 1Ø, STATUS = 'OLD', BLANK =
     & 'ZERO', READONLY)
      WRITE (6, 15) 'PLOT STATION NAMES (Y OR N)?  '
2Ø    READ (5, 16, IOSTAT = IOS) ANS
      IF ((ANS .NE. 'Y' .AND. ANS .NE. 'N') .OR. IOS .NE. Ø) THEN
        PRINT *, '***** PLEASE ANSWER "Y" OR "N"; TRY AGAIN *****'
        GOTO 2Ø
      END IF
      WRITE (6, 15) 'ENTER NUMBER OF MECHANISMS TO SKIP (INCLUDING MULTI
     &PLE SOLUTIONS):  '
25    READ (5, *, IOSTAT = IOS) NSKIP
      IF (NSKIP .LT. Ø .OR. IOS .NE. Ø) THEN
        PRINT *, '***** INVALID NUMBER; TRY AGAIN *****'
        GOTO 25
      END IF
C
C READ HYPO71 HEADER CARD (FIRST LINE IN MODEL FILE)
C
      NEV = Ø
      ILINE = 1
      READ (2, 3Ø, ERR = 2ØØØ) TITLE
3Ø    FORMAT (A)
      INDX = 1
      DO 4Ø I = 1, 8Ø
      IF (TITLE(I:I) .NE. ' ') THEN
        INDX = I
        GOTO 5Ø
      END IF
4Ø    CONTINUE
5Ø    TITLE = TITLE(INDX:LEN(TITLE))
C
C INITIALIZE PLOT PROGRAM
C
      CALL PLOTS (Ø., Ø., Ø)
C
C READ EVENT
C
6Ø      ILINE = ILINE + 1
        READ (2, 7Ø, END = 1ØØØ, ERR = 2ØØØ) EVENT
7Ø      FORMAT (1X, A132)
        NEV = NEV + 1
        IF (NEV .GT. NSKIP) THEN
C
C START NEW FRAME
C
          CALL ERASE
          CALL DELAY ('ØØØØ ØØ:ØØ:Ø1.ØØ')
          CALL PLOT (1., 1., -3)
          FIRST = .TRUE.
          CALL NEWPEN (1)
          READ (EVENT, 75) DD1, DA1, SA1
75        FORMAT (T81, F4.Ø, F3.Ø, F4.Ø)
C
C PLOT SUMMARY CARD & EXPLANATION OF SYMBOLS
C
          YPOS = YPOS1
          CALL SYMBOL (XPOS1, YPOS, HITE3, %REF(TITLE), Ø., LEN(TITLE))
          YPOS =YPOS - .3
          CALL SYMBOL (XPOS1, YPOS, HITE3, %REF(EVENT(1:8Ø)), Ø., 8Ø)
          YPOS =YPOS - .3
```

```
          CALL SYMBOL (XPOS1, YPOS, HITE3, XREF(EVENT(82:132)), 0., 51)
          CALL NEWPEN (1)
          YPOS = YPOS2
          CALL CIRCLE (HITE2, 2.0*PI, XPOS1, YPOS + HITE2/2.)
          LINE(1:11) = 'COMPRESSION'
          CALL SYMBOL(XPOS1 + .2, YPOS, HITE2, XREF(LINE), 0.,11)

          YPOS = YPOS + 2.0*HITE2
          CALL TRINGL (HITE2, XPOS1, YPOS + HITE2/2.)
          LINE(1:11) = 'DILATATION '
          CALL SYMBOL(XPOS1 + .2, YPOS, HITE2, XREF(LINE),0.,11)
C
C PLOT BIG & LITTLE STEREO NET PERIMETERS
C
          CALL STRNET (CX1, CY1, RAD, RMAX1)
          CALL STRNET (CX2, CY2, RAD, RMAX2)
C
C PLOT NODAL PLANES
C
          CALL PLOTPL (CX1, CY1, DA1, PI, RAD, RMAX1, DD1 - 90.)
          CALL AUXPLN (DD1, DA1, SA1, DD2, DA2, SA2)
          CALL PLOTPL (CX1, CY1, DA2, PI, RAD, RMAX1, DD2 - 90.)
C
C  PLOT "P" AND "T" AXES IN BIG NET
C
          NAME = '    '
          WT = 1.0
          CALL NEWPEN (3)
          CALL TPPLOT (CX1, CY1, DA1, DD1, HITE1, PI, RAD, RMAX1, SA1,
     & WT)
C
C PLOT THE P AND T AXES IN SMALL NET CORRESPONDING TO THE SET OF "NEIGHBORING SOLUTIONS"
C
          CALL TPPLOT (CX2, CY2, DA1, DD1, HITE2, PI, RAD, RMAX2, SA1,
     & WT)
          CALL NEWPEN (1)
        END IF
        ILINE = ILINE + 1
        READ (2, *, ERR = 2000) NSTAR
        NLINE = (NSTAR-1)/11 + 1
        DO 90 I = 1, NLINE
          IF (I .LT. NLINE) THEN
            NREAD = 11
          ELSE
            NREAD = MOD(NSTAR, 11)
          END IF
          ILINE = ILINE + 1
          IF (NEV .GT. NSKIP) THEN
            READ (2, 80, ERR = 2000) (DDN(N), DAN(N), SAN(N), N = 1,
     & NREAD)
80          FORMAT (1X, 11(F4.0, F3.0, F4.0))
            DO 85 N = 1, NREAD
              CALL TPPLOT (CX2, CY2, DAN(N), DDN(N), HITE2, PI, RAD,
     & RMAX2, SAN(N), WT)
85          CONTINUE
          END IF
90      CONTINUE
C
C READ PHASE CARD
C
100     ILINE = ILINE + 1
        READ (2, 110, END = 1000, ERR = 2000) NAME, DIST, AZM, AIN,
     & PRMK, WT, DISC
110     FORMAT (1X, A4, 3F6.1, 3X, A4, F5.2, 2X, A1)
        IF (NAME .NE. '    ') THEN
C
C REPORT DISCREPANT OBSERVATIONS
C
          IF (NEV .GT. NSKIP) THEN
            IF (DISC .EQ. '*') THEN
              IF (FIRST) THEN
                LINE(1:23) = 'DISCREPANT OBSERVATIONS'
                YPOS = YPOS3
                CALL SYMBOL (XPOS2, YPOS, HITE2, XREF(LINE), 0., 23)
                YPOS = YPOS - HITE2*1.5
```

```
                  LINE(1:25) = 'STAT DIST  AZM  AIN  PRMK'
                  CALL SYMBOL (XPOS2, YPOS. HITE2. %REF(LINE), Ø., 25)
                  YPOS = YPOS - HITE2*.1
                  LINE(1:25) = '_________________________'
                  CALL SYMBOL (XPOS2, YPOS. HITE2. %REF(LINE), Ø., 25)
                  YPOS = YPOS - HITE2*1.5
                  FIRST = .FALSE.
                END IF
                WRITE (LINE, 12Ø) NAME, DIST, NINT(AZM), NINT(AIN), PRMK
12Ø             FORMAT (A4, F5.1, 2I5, 2X, A4, I5X)
                CALL SYMBOL (XPOS2, YPOS, HITE2, %REF(LINE), Ø., 31)
                YPOS = YPOS - HITE2*1.5
C
C PLOT FIRST MOTION
C
              END IF
              IF (ANS .EQ. 'N') NAME = '    '
              IF (PRMK(3:3) .EQ. 'U' .OR. PRMK(3:3) .EQ. '+') THEN
                SYM = 'C'
              ELSE
                SYM = 'D'
              END IF
              CALL PLTSYM (AIN, AZM, CX1, CY1, HITE1, NAME, PI, RAD,
     & RMAX1, SYM, WT)
            END IF
            GOTO 1ØØ
          END IF
          IF (NEV .GT. NSKIP) CALL PLOT (1., 1., -999)
        GOTO 6Ø
C
C END OF FILE
C
1ØØØ    CALL PLOT (Ø., Ø., 999)
        CLOSE (2)
        STOP
C
C READ ERROR
C
2ØØØ    PRINT *, 'READ ERROR ON LINE ', ILINE
        STOP
        END
C
C
C
C
C
        SUBROUTINE ERASE
C
C SENDS A SCREEN ERASE CODE TO A TEKTRONIX TERMINAL
C
        CHARACTER*1       A                         ! ASCII ESCAPE
        CHARACTER*1       B                         ! ASCII FORM FEED
C
        A = CHAR(27)
        B = CHAR(12)
        WRITE (6, 1Ø) A, B
1Ø      FORMAT (1X, 2A1)
C
        RETURN
        END
C
C
C
C
```

```
C
        SUBROUTINE DELAY (TIME)
C
C CREATES A DELAY FOR SCREEN RECOVERY
C
      CHARACTER*(*)      TIME                    ! STANDARD VMS TIME FORMAT OF THE FORM 'DDDD HH:MM:SS.SS'
      DOUBLE PRECISION   B                       ! BINARY TIME CORRESPONDING TO TIME
C
C CONVERT ASCII TIME TO BINARY TIME
C
      CALL SYS$BINTIM (TIME,B)
C
C SCHEDULE A WAKEUP FOR A ELAPSED TIME (NEGATIVE B)
C
      CALL SYS$SCHDWK (,,-B,)
C
C HIBERNATE AND REAWAKE
C
      CALL SYS$HIBER
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE CIRCLE (SIZE, TWOPI, XØ, YØ)
C
C PLOT A CIRCLE
C
      REAL               SIZE                    ! SIZE OF CIRCLE
      REAL               TWOPI                   ! TWO*PI
      REAL               XØ                      ! X POSTION OF CENTER
      REAL               YØ                      ! Y POSTION OF CENTER
C
      REAL               ANGLE                   ! ANGLE
      INTEGER            J                       ! LOOP INDEX
      INTEGER            N                       ! NUMBER OF POINTS OF WHICH CIRCLE COMPOSED
      REAL               SIZE2                   ! SCRATCH VARIABLE
      REAL               X                       ! X PLOT POSTION
      REAL               Y                       ! Y PLOT POSTION
C
      SIZE2 = SIZE*Ø.5
C
C COMPUTE OPTIMUM # OF POINTS TO DRAW
C
      N = 2Ø*SQRT(SIZE2*2Ø.)
      IF (N .LT. 1Ø) N = 1Ø
C
C DRAW CIRCLE
C
      X = XØ + SIZE2
      CALL PLOT (X, YØ, 3)
      DO 1Ø J = 1, N
        ANGLE = TWOPI*FLOAT(J)/FLOAT(N)
        X = XØ + SIZE2*COS(ANGLE)
        Y = YØ + SIZE2*SIN(ANGLE)
        CALL PLOT (X, Y, 2)
1Ø    CONTINUE
C
      RETURN
      END
C
```

```
C
C
C
C
      SUBROUTINE TRINGL (SIZE, XØ, YØ)
C
C PLOT A TRIANGE
C
      REAL              SIZE                       ! SIZE OF TRIANGE
      REAL              XØ                         ! X POSTION OF CENTER
      REAL              YØ                         ! Y POSTION OF CENTER
C
      REAL              SIZEX                      ! SCRATCH VARIABLE
      REAL              SIZEY                      ! SCRATCH VARIABLE
      REAL              X                          ! X PLOT POSTION
      REAL              Y                          ! Y PLOT POSTION
C
      PARAMETER (COS3Ø = Ø.866Ø254, SIN3Ø = Ø.5)
C
      SIZEY = SIZE*SIN3Ø
      SIZEX =  SIZEY*COS3Ø
C
C MOVE TO TOP
C
      Y = YØ + SIZEY
      CALL PLOT (XØ, Y, 3)
C
C DRAW TO LOWER LEFT
C
      X = XØ - SIZEX
      Y = YØ - SIZEY/2
      CALL PLOT (X, Y, 2)
C
C DRAW TO LOWER RIGHT
C
      X = XØ + SIZEX
      CALL PLOT (X, Y, 2)
C
C DRAW TO TOP
C
      Y = YØ + SIZEY
      CALL PLOT (XØ, Y, 2)
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE STRNET (CX, CY, RAD, RMAX)
C
C PLOT PERIMETER OF A STEREO NET
C
      REAL              CX                         ! X POSITION OF CIRCLE CENTER
      REAL              CY                         ! Y POSITION OF CIRCLE CENTER
      REAL              RAD                        ! PI/18Ø
      REAL              RMAX                       ! RADIUS OF CIRCLE

      REAL              CSIZ                       ! SCRATCH VARIABLE (RMAX/1ØØ)
      INTEGER           I                          ! LOOP INDEX OVER DEGREES
      INTEGER           N                          ! TESTS 1Ø DEGREE TICK POSITION
      INTEGER           NN                         ! TESTS 9Ø DEGREE TICK POSITION
      REAL              P                          ! TICK LENGTH
```

```
      REAL              PHI                           ! AZIMUTH IN RADIANS
      REAL              X                             ! X POSTION OF CIRCLE
      REAL              XP                            ! X POSITION OF END OF TICK
      REAL              Y                             ! Y POSTION OF CIRCLE
      REAL              YP                            ! Y POSITION OF END OF TICK
C
      CALL NEWPEN (2)
C
C DRAW CIRCLE @ 1 DEGREE INCREMENTS
C
      DO 1Ø I = 1, 361
        PHI = (I - 1)*RAD
        X = RMAX*COS(PHI) + CX
        Y = RMAX*SIN(PHI) + CY
        N = (I - 1) - ((I - 1)/1Ø)*1Ø
        NN = (I - 1) - ((I - 1)/9Ø)*9Ø
        P = Ø.Ø1*RMAX
        IF ((N .EQ. Ø) .AND. (I .GT. 1Ø)) P = Ø.Ø2*RMAX
        IF ((NN .EQ. Ø) .AND. (I .GT. 9Ø)) P = Ø.Ø4*RMAX
        XP = (RMAX + P)*COS(PHI) + CX
        YP = (RMAX + P)*SIN(PHI) + CY
        IF (I .GT. 1) THEN
          CALL PLOT (X, Y, 2)
          CALL PLOT (XP, YP, 2)
          CALL PLOT (X, Y, 2)
        ELSE
          CALL PLOT (X, Y, 3)
        END IF
1Ø    CONTINUE
C
C PLOT + AT CENTER
C
      CSIZ = .Ø1*RMAX
      CALL PLOT (CX - CSIZ, CY, 3)
      CALL PLOT (CX + CSIZ, CY, 2)
      CALL PLOT (CX, CY - CSIZ, 3)
      CALL PLOT (CX, CY + CSIZ, 2)
      CALL NEWPEN (1)
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE PLOTPL (CX, CY, DPIDG, PI, RAD, RMAX, STRKDG)
C
C PLOTS FAULT PLANE ON LOWER HEMISPHERE STEREO NET
C
      REAL              CX                            ! X POSITION OF CIRCLE CENTER
      REAL              CY                            ! Y POSITION OF CIRCLE CENTER
      REAL              DPIDG                         ! DIP ANGLE IN DEGREES
      REAL              PI                            ! PI
      REAL              RAD                           ! PI/18Ø
      REAL              RMAX                          ! RADIUS OF CIRCLE
      REAL              STRKDG                        ! STRIKE ANGLE IN DEGREES

      REAL              ANG                           ! ANGLE IN RADIANS
      REAL              AINP(91)                      ! ANGLE OF INCIDENCE IN RADIANS
      REAL              ARG                           ! DUMMY ARGUMENT
      REAL              AZ                            ! AZIMUTH
      REAL              CON                           ! RADIUS COEFFICIENT
      REAL              DIPRD                         ! DIP ANGLE IN RADIANS
```

```
      INTEGER          I                              ! LOOP INDEX
      INTEGER          MI                             ! SCRATCH INDEX
      REAL             RADIUS                         ! RADIUS
      REAL             SAZ(91)                        ! AZIMUTH IN RADIANS
      REAL             STRKRD                         ! STRIKE IN RADIANS
      REAL             TAZ                            ! SCRATCH VARIABLE
      REAL             TPD                            ! SCRATCH VARIABLE
      REAL             X                              ! X PLOT POSITION
      REAL             Y                              ! Y PLOT POSITION
C
      STRKRD = STRKDG*RAD
      DIPRD = DPIDG*RAD
      TPD = TAN(PI*.5 - DIPRD)**2
C
C CASE OF VERTICAL PLANE
C
      IF (DPIDG .EQ. 9Ø.Ø) THEN
        X = RMAX*SIN(STRKRD) + CX
        Y = RMAX*COS(STRKRD) + CY
        CALL PLOT (X, Y, 3)
        X = RMAX*SIN(STRKRD + PI) + CX
        Y = RMAX*COS(STRKRD + PI) + CY
        CALL PLOT (X, Y, 2)
        RETURN
      END IF
C
C COMPUTE ANGLE OF INCIDENCE, AZIMUTH
C
      DO 1Ø I = 1, 9Ø
        ANG = FLOAT(I - 1)*RAD
        ARG = SQRT((COS(DIPRD)**2)*(SIN(ANG)**2))/COS(ANG)
        SAZ(I) = ATAN(ARG)
        TAZ = TAN(SAZ(I))**2
        ARG = SQRT(TPD + TPD*TAZ + TAZ)
        AINP(I) = ACOS(TAN(SAZ(I))/ARG)
   1Ø CONTINUE
      SAZ(91) = 9Ø.*RAD
      AINP(91) = PI*.5 - DIPRD
C
C PLOT PLANE
C
      CON = RMAX*SQRT(2.)
      DO 2Ø I = 1, 18Ø
        IF (I .LE. 91) THEN
          MI = I
          AZ = SAZ(I) + STRKRD
        ELSE
          MI = 181 - I
          AZ = PI - SAZ(MI) + STRKRD
        END IF
        RADIUS = CON*SIN(AINP(MI)*Ø.5)
        X = RADIUS*SIN(AZ) + CX
        Y = RADIUS*COS(AZ) + CY
        IF (I .EQ. 1) THEN
          CALL PLOT (X, Y, 3)
        ELSE
          CALL PLOT (X, Y, 2)
        END IF
 2Ø   CONTINUE
C
      RETURN
      END
C
C
```

```
C
C
C
      SUBROUTINE AUXPLN (DD1, DA1, SA1, DD2, DA2, SA2)
C
C     CALCULATE THE AUXILLIARY PLANE OF A DOUBLE COUPLE FAULT PLANE SOLUTION, GIVEN THE PRINCIPLE PLANE.
C
C     WRITTEN BY PAUL REASENBERG, JUNE, 1984, FROM CLASS NOTES BY DAVE BOORE, (BOTH AT THE U.S.G.S., MENLO PARK.)
C     ANGLE VARIABLES PHI, DEL AND LAM ARE AS DEFINED IN AKI AND RICHARDS, (1980), P.114.
C
      REAL              DA1                              ! DIP ANGLE IN DEGREES
      REAL              DD1                              ! DIP DIRECTIONS IN DEGREES
      REAL              SA1                              ! SLIP ANGLE IN DEGREES
      REAL              DA2                              ! DIP ANGLE OF AUXILLIARY PLANE
      REAL              DD2                              ! DIP DIRECTION OF AUXILLIARY PLANE
      REAL              SA2                              ! SLIP ANGLE OF AUXILLARY PLANE
C

      DOUBLE PRECISION  BOT                              ! SCRATCH VARIABLE
      DOUBLE PRECISION  DEL1                             ! DIP ANGLE OF PRINCIPAL PLANE IN RADIANS
      LOGICAL           FIRST                            ! TEST: TRUE IF FIRST TIME INTO ROUTINE
      DOUBLE PRECISION  PHI1                             ! FAULT PLANE STRIKE OF PRINCIPAL PLANE
      DOUBLE PRECISION  PHI2                             ! STRIKE OF AUXILLIARY PLANE IN RADIANS
      DOUBLE PRECISION  RAD                              ! CONVERSION FACTOR FROM DEGREES TO RADIAN
      DOUBLE PRECISION  SGN                              ! SAVES PRINCIPAL PLANE SLIP ANGLE FOR ASSIGNING PROPER SIGN TO AUXILLIARY
      DOUBLE PRECISION  TOP                              ! SCRATCH VARIABLE
      DOUBLE PRECISION  XLAM1                            ! SLIP ANGLE OF PRINCIPAL PLANE IN RADIANS
      DOUBLE PRECISION  XLAM2                            ! SLIP ANGLE OF AUXILLIARY PLANE
C
      DATA FIRST /.TRUE./
      SAVE FIRST, RAD
C
      IF (FIRST) THEN
        FIRST = .FALSE.
        RAD = DATAN(1.0D0)/45.0D0
      END IF
C
      PHI1 = DD1 - 90.0D0
      IF (PHI1 .LT. 0.0D0) PHI1 = PHI1 + 360.0D0
      PHI1 = PHI1*RAD
      DEL1 = DA1*RAD
      SGN = SA1
      XLAM1 = SA1*RAD
C
      TOP = DCOS(XLAM1)*DSIN(PHI1) - DCOS(DEL1)*DSIN(XLAM1)*DCOS(PHI1)
      BOT = DCOS(XLAM1)*DCOS(PHI1) + DCOS(DEL1)*DSIN(XLAM1)*DSIN(PHI1)
      DD2 = DATAN2(TOP, BOT)/RAD
      PHI2 = (DD2 - 90.0D0)*RAD
      IF (SA1 .LT. 0.0D0) DD2 = DD2 - 180.0D0
      IF (DD2 .LT. 0.0D0) DD2 = DD2 + 360.0D0
      IF (DD2. GT. 360.0D0) DD2 = DD2 - 360.0D0
C
      DA2 = DACOS(DSIN(DABS(XLAM1))*DSIN(DEL1))/RAD
      XLAM2 = -DCOS(PHI2)*DSIN(DEL1)*DSIN(PHI1) +
     & DSIN(PHI2)*DSIN(DEL1)*DCOS(PHI1)
C
C MACHINE ACCURACY PROBLEM
C
      IF (XLAM2 .GT. 1.0D0) THEN
        XLAM2 = 1.0D0
      END IF
      XLAM2 = DSIGN(DACOS(XLAM2), SGN)
      SA2 = XLAM2/RAD
C
```

```
      RETURN
      END
C
C
C
C
C
      SUBROUTINE TPPLOT (CX, CY, DA1, DD1, HITE, PI, RAD, RMAX, SA1, WT)
C
C PLOT P AND T AXES
C
      REAL              CX                              ! X POSITION OF CIRCLE CENTER
      REAL              CY                              ! Y POSITION OF CIRCLE CENTER
      REAL              DA1                             ! DIP ANGLE OF PRINCIPLE PLANE
      REAL              DD1                             ! DIP DIRECTION OF PRINCIPLE PLANE
      REAL              HITE                            ! HEIGHT OF P,T SYMBOL
      REAL              PI                              ! PI
      REAL              RAD                             ! PI/18Ø
      REAL              RMAX                            ! RADIUS OF CIRCLE
      REAL              SA1                             ! RAKE OF PRINCIPLE PLANE
      REAL              WT                              ! WEIGHT ASSIGNED TO PICK QUALITY IN PROGRAM FPFIT
C
      REAL              AIN1                            ! ANGLE OF INCIDENCE OF P/T AXIS
      REAL              AIN2                            ! ANGLE OF INCIDENCE OF T/P AXIS
      REAL              ANG                             ! ANGLE OF PLOT SYMBOL
      REAL              AZ1                             ! AZIMUTH OF P/T AXIS
      REAL              AZ2                             ! AZIMUTH OF T/P AXIS
      CHARACTER*4       BLANK                           ! BLANK
      REAL              DA2                             ! DIP ANGLE OF AUXILLIARY PLANE
      REAL              DD2                             ! DIP DIRECTION OF AUXILLIARY PLANE
      REAL              SA2                             ! STRIKE OF AUXILLIARY PLANE
      CHARACTER*1       SYM1                            ! P/T PLOT SYMBOL
      CHARACTER*1       SYM2                            ! T/P PLOT SYMBOL
C
      PARAMETER (ANG = Ø.Ø, BLANK = '    ')
C
C FIND AUXILLIARY PLANE
C
      CALL AUXPLN (DD1, DA1, SA1, DD2, DA2, SA2)
C
C FIND P AND T AXES
C
      CALL TANDP (AIN1, AIN2, AZ1, AZ2, DA1, DA2, DD1, DD2, PI, RAD)
      IF (SA1 .LT. Ø.) THEN
        SYM1 = 'P'
        SYM2 = 'T'
      ELSE
        SYM1 = 'T'
        SYM2 = 'P'
      END IF
C
C PLOT SYMBOLS
C
      CALL PLTSYM (AIN1, AZ1, CX, CY, HITE, BLANK, PI, RAD, RMAX, SYM1,
     & WT)
      CALL PLTSYM (AIN2, AZ2, CX, CY, HITE, BLANK, PI, RAD, RMAX, SYM2,
     & WT)
C
      RETURN
      END
C
C
C
C
```

```
C
      SUBROUTINE TANDP(AIN1, AIN2, AZ1, AZ2, DA1, DA2, DD1, DD2, PI,RAD)
C
C     GIVEN TWO PLANES COMPUTE AZ AND ANGLE OF INCIDENCE OF P & T AXES
C
      REAL              AIN1                          ! ANGLE OF INCIDENCE OF P/T AXIS
      REAL              AIN2                          ! ANGLE OF INCIDENCE OF T/P AXIS
      REAL              AZ1                           ! AZIMUTH OF P/T AXIS
      REAL              AZ2                           ! AZIMUTH OF T/P AXIS
      REAL              DA1                           ! DIP ANGLE OF PRINICIPLE PLANE
      REAL              DA2                           ! DIP ANGLE OF AUXILLIARY PLANE
      REAL              DD1                           ! DIP DIRECTION OF PRINCIPLE PLANE
      REAL              DD2                           ! DIP DIRECTION OF AUXILLIARY PLANE
      REAL              PI                            ! PI
      REAL              RAD                           ! PI/18Ø
C
      REAL              ALAT1                         ! DIP ANGLE IN RADIANS OF PRINCIPLE PLANE MEASURED FROM VERTICAL
      REAL              ALAT2                         ! DIP ANGLE IN RADIANS OF AUXILLIARY PLANE MEASURED FROM VERTICAL
      REAL              ALON1                         ! DD1 IN RADIANS
      REAL              ALON2                         ! DD2 IN RADIANS
      REAL              AZIMTH                        ! AZIMUTH IN RADIANS OF POLE ??
      REAL              AZØ                           ! AZIMUTH FROM POLE OF AUXILLIARY PLANE TO POLE OF PRINCIPLE ??
      REAL              BAZM                          ! NOT USED
      REAL              DELTA                         ! NOT USED
      REAL              PLUNGE                        ! PLUNGE IN RADIANS OF POLE ??
      REAL              SHIFT                         ! AZIMUTHAL SHIFT FROM POLE OF PLANE TO P TO T AXIS (= 45 DEGREES)??
      REAL              XPOS                          ! NOT USED
      REAL              YPOS                          ! NOT USED
C
      PARAMETER (SHIFT = Ø.7853981)
C
      ALAT1 = (9Ø. - DA1)*RAD
      ALON1 = DD1*RAD
      ALAT2 = (9Ø. - DA2)*RAD
      ALON2 = DD2*RAD
      CALL REFPT (ALAT2, ALON2)
      CALL DELAZ (ALAT1, ALON1, DELTA, AZØ, BAZM, XPOS, YPOS)
      CALL BACK (SHIFT, AZØ, PLUNGE, AZIMTH)
      IF (ABS(AZIMTH) .GT. PI) AZIMTH = AZIMTH - SIGN(2.Ø*PI, AZIMTH)
      AZ1 = AZIMTH/RAD
      AIN1 = PLUNGE/RAD + 9Ø.
      AZØ = AZØ + PI
      CALL BACK (SHIFT, AZØ, PLUNGE, AZIMTH)
      IF (ABS(AZIMTH) .GT. PI) AZIMTH = AZIMTH - SIGN(2.Ø*PI, AZIMTH)
      AZ2 = AZIMTH/RAD
      AIN2 = PLUNGE/RAD + 9Ø.
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE GEOCEN
C
C      GEOCEN - CALCULATE GEOCENTRIC POSTITIONS, DISTANCES, AND AZIMUTHS (BRUCE JULIAN, USGS MENLO PARK, CA)
C
C      THE GEOCENTRIC DISTANCE DELTA AND AZIMUTH AZØ FROM POINT (LATØ, LONØ) TO POINT (LAT1, LON1) ARE CALCULTED FROM
C            COS(DELTA) = COS(LATØ')*COS(LAT1')*COS(LON1 - LONØ) + SIN(LATØ')*SIN(LAT1')
C            SIN(DELTA) = SQRT(A*A + B*B)
C            TAN(AZØ) = A/B
C
C      WHERE
```

```
C           A = COS(LAT1')*SIN(LON1-LONØ)
C           B = COS(LATNØ')*SIN(LAT1') - SIN(LATØ')*COS(LAT1')*COS(LON1 - LONØ)
C           LATØ', LAT1' = GEOCENTRIC LATITUDES OF POINTS
C           LONØ, LON1 = LONGITUDES OF POINTS
C
C      THE GEOCENTRIC LATITUDE LAT' IS GOTTEN FROM THE GEOGRAPHIC LATITUDE LAT BY TAN(LAT') = (1 - ALPHA)*(1 - ALPHA)*TAN(LAT),
C      WHERE ALPHA IS THE FLATTENING OF THE ELLIPSOID.  SEE FUNCTION GGTOGC FOR CONVERSION.
C      THE BACK AZIMUTH IS CALCULATED BY THE SAME FORMULAS WITH (LATØ', LONØ) AND (LAT1', LON1) INTERCHANGED.
C      AZIMUTH IS MEASURED CLOCKWISE FROM NORTH THRU EAST.
C
      REAL              R, THETA                      ! RADIUS, AZIMUTH IN POLAR COORDINATES
      REAL              AZØ                           ! AZIMUTH FROM REFERENCE POINT TO SECONDARY POINT IN RADIANS
      REAL              AZ1                           ! AZIMUTH FROM SECONDARY POINT TO REFERENCE POINT IN RADIANS
      REAL              CDELT                         ! SINE OF DELTA TO SECONDARY POINT
      REAL              CDLON                         ! COSINE OF DIFFERENCE OF SECONDARY POINT, REFERENCE LONGITUDE
      REAL              COLAT                         ! AVERAGE COLATITUDE OF STATION
      REAL              CTØ                           ! SINE OF REFERENCE POINT LATITUDE
      REAL              CT1                           ! SINE OF SECONDARY POINT LATITUDE
      REAL              CZØ                           ! COSINE OF AZIMUTH TO SECONDARY POINT
      REAL              DELTA                         ! GEOCENTRIC DISTANCE IN DEGREES
      REAL              DLON                          ! AZIMUTH IN POLAR COORDINATES TO SECONDARY POINT ?
      REAL              ERAD                          ! EQUATORIAL RADIUS (CHOVITZ, 1981, EOS, 62, 65-67)
      REAL              FLAT                          ! EARTH FLATTENING CONSTANT (CHOVITZ, 1981, EOS, 62, 65-67)
      INTEGER           LAMBDA                        ! DUMMY VARIABLE
      REAL              LAT                           ! LATITUDE IN RADIANS
      REAL              LON                           ! LONGITUDE IN RADIANS
      REAL              OLAT                          ! ORIGIN LATITUDE IN RADIANS
      REAL              OLON                          ! ORIGIN LONGITUDE IN RADIANS
      REAL              PHIØ                          ! REFERENCE SECONDARY POINT LONGITUDE
      REAL              PI                            ! 3.14159...
      REAL              RADIUS                        ! EARTH RADIUS AT COLAT
      REAL              SDELT                         ! COSINE OF DELTA TO SECONDARY POINT
      REAL              SDLON                         ! SINE OF DIFFERENCE OF SECONDARY POINT, REFERENCE LONGITUDE
      REAL              STØ                           ! COSINE OF REFERENCE POINT LATITUDE
      REAL              ST1                           ! COSINE OF SECONDARY POINT LATITUDE
      REAL              TWOPI                         ! 2*PI
C
      SAVE STØ, CTØ, PHIØ, OLAT
      PARAMETER (PI = 3.1415926535897, TWOPI = 2.*PI)
      PARAMETER (FLAT = 1./298.257, ERAD = 6378.137)
      PARAMETER (LAMBDA = FLAT*(2. - FLAT)/(1. - FLAT)**2)
C
C REFPT - STORE THE GEOCENTRIC COORDINATES OF THE REFEERNCE POINT
C
      ENTRY REFPT(OLAT, OLON)
C
      STØ = COS(OLAT)
      CTØ = SIN(OLAT)
      PHIØ = OLON
      RETURN
C
C DELAZ - CALCULATE THE GEOCENTRIC DISTANCE, AZIMUTHS
C
      ENTRY DELAZ(LAT, LON, DELTA, AZØ, AZ1, X, Y)
C
      CT1 = SIN(LAT)
      ST1 = COS(LAT)
      IF ((CT1 - CTØ) .EQ. Ø. .AND. (LON - PHIØ) .EQ. Ø.) THEN
        DELTA = Ø.
        AZØ = Ø.
        AZ1 = Ø.
      ELSE
        SDLON = SIN(LON - PHIØ)
        CDLON = COS(LON - PHIØ)
```

```
        CDELT = STØ*ST1*CDLON + CTØ*CT1
        CALL CVRTOP (STØ*CT1 - ST1*CTØ*CDLON, ST1*SDLON, SDELT, AZØ)
        DELTA = ATAN2(SDELT, CDELT)
        CALL CVRTOP (ST1*CTØ - STØ*CT1*CDLON, -SDLON*STØ, SDELT, AZ1)
        IF (AZØ .LT. Ø.Ø) AZØ = AZØ + TWOPI
        IF (AZ1 .LT. Ø.Ø) AZ1 = AZ1 + TWOPI
      END IF
      COLAT = PI/2. - (LAT + OLAT)/2.
      RADIUS = ERAD/SQRT(1.Ø + LAMBDA*COS(COLAT)**2)
      Y = RADIUS*DELTA*COS(AZØ)
      X = RADIUS*DELTA*SIN(AZØ)
      RETURN
C
C BACK - CALCULATE GEOCENTRIC COORDINATES OF SECONDARY POINT FROM DELTA, AZ
C
      ENTRY BACK (DELTA, AZØ, LAT, LON)
C
      SDELT = SIN(DELTA)
      CDELT = COS(DELTA)
      CZØ = COS(AZØ)
      CT1 = STØ*SDELT*CZØ + CTØ*CDELT
      CALL CVRTOP (STØ*CDELT - CTØ*SDELT*CZØ, SDELT*SIN(AZØ), ST1, DLON)
      LAT = ATAN2(CT1, ST1)
      LON = PHIØ + DLON
      IF (ABS(LON) .GT. PI) LON = LON - SIGN(TWOPI, LON)
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE CVRTOP(X, Y, R, THETA)
C
C CVRTOP - CONVERT FROM RECTANGULAR TO POLAR COORDINATES (BRUCE JULIAN, USGS MENLO PARK, CA)
C
      REAL              X,Y                            ! X,Y RECTANGULAR COORDINATES
      REAL              R, THETA                       ! RADIUS, AZIMUTH IN POLAR COORDINATES
C
      R = SQRT(X*X + Y*Y)
      THETA = ATAN2(Y, X)
      RETURN
      END
C
C
C
C
C
      SUBROUTINE PLTSYM (AIN, AZ, CX, CY, HITE, NAME, PI, RAD, RMAX,
     & SYM, WT)
C
C    PLOT EITHER FIRST MOTION SYMBOL (C,D,+,-) WITH STATION NAME NEXT TO SYMBOL, OR STRESS AXES SYMBOL (P & T)
C
      REAL              AIN                            ! ANGLE OF INCIDENCE OF SYMBOL
      REAL              AZ                             ! AZIMUTH OF SYMBOL
      REAL              CX                             ! X POSITION OF CIRCLE CENTER
      REAL              CY                             ! Y POSITION OF CIRCLE CENTER
      REAL              HITE                           ! HEIGHT OF SYMBOL
      CHARACTER*4       NAME                           ! STRING TO BE PLOTTED TO RIGHT OF SYMBOL
      REAL              PI                             ! PI
      REAL              RAD                            ! PI/18Ø
      REAL              RMAX                           ! RADIUS OF CIRCLE
      CHARACTER*1       SYM                            ! PLOT SYMBOL
```

```
      REAL            WT                           ! WEIGHT ASSIGNED TO PICK QUALITY IN PROGRAM FPFIT

      REAL            AINR                         ! AIN IN RADIANS
      REAL            ANG                          ! PLOT ANGLE OF SYMBOL
      REAL            AZR                          ! AZ IN RADIANS
      REAL            CON                          ! RMAX * SQRT(2.Ø)
      LOGICAL         FIRST                        ! FLAG FIRST TIME INTO ROUTINE
      REAL            R                            ! DISTANCE FROM CX, CY TO PLOT POSITION
      REAL            SIZE                         ! PLOT SYMBOL SIZE SCALED BY WT
      REAL            SYMSIZ                       ! MAXIMUM SYMBOL SIZE
      REAL            X                            ! X POSITION OF SYMBOL
      REAL            Y                            ! Y POSITION OF SYMBOL
C
      PARAMETER (ANG = Ø.Ø, SYMSIZ = Ø.2)
C
      AZR = AZ*RAD
      AINR = AIN*RAD
C
C UPGOING RAYS
C
      IF (AIN .GT. 9Ø.) THEN
        AINR = PI - AINR
        AZR = PI + AZR
      END IF
      CON = RMAX*SQRT(2.Ø)
      R = CON*SIN(AINR*Ø.5)
      X = R*SIN(AZR) + CX
      Y = R*COS(AZR) + CY
C
C STRESS AXIS SYMBOL
C
      IF ((SYM .EQ. 'P') .OR. (SYM .EQ. 'T')) THEN
        X = X - .286*HITE
        Y = Y - .5*HITE
        CALL SYMBOL (X, Y, HITE, %REF(SYM), ANG, 1)
      ELSE
C
C FIRST MOTION SYMBOL
C
        SIZE = SYMSIZ*WT
        IF (AIN .GT. 9Ø.) THEN
          CALL NEWPEN (4)
        ELSE
          CALL NEWPEN (1)
        END IF
        IF (SYM .EQ. 'C') THEN
          CALL CIRCLE (SIZE, 2.Ø*PI, X, Y)
        ELSE
          CALL TRINGL (SIZE, X, Y)
        END IF
        CALL NEWPEN (1)
C
C PLOT STATION NAME
C
        IF (NAME .NE. '    ') CALL SYMBOL (X + SIZE/2., Y, SIZE/2.,
     & %REF(NAME), Ø., 4)
      END IF
C
      RETURN
      END
```

```
      PROGRAM FPPAGE
C
C     PURPOSE:        PLOT EARTHQUAKE RAY POLARITIES AND FAULT PLANES ON A LOWER HEMISPHERE EQUAL AREA PROJECTION.
C                     MAKES MULTIPLE PLOTS PER PAGE.
C
C     INPUT FILE:     A FILE OF THE TYPE "*.POL", WHICH IS GENERATED BY THE PROGRAM "FPFIT" (SEE "FPFIT, FPPLOT AND FPPAGE:
C                     FORTRAN COMPUTER PROGRAMS FOR CALCULATING AND DISPLAYING EARTHQUAKE FAULT-PLANE SOLUTIONS,
C                     BY P. REASENBERG AND D. OPPENHEIMER, U.S. GEOLOGICAL SURVEY OPEN-FILE REPORT 85-???)
C
C     REQUIRED ROUTINES: CALCOMP STYLE PLOT ROUTINES PLOTS, PLOT, NEWPEN, SYMBOL.
C
C     DEPARTURES FROM FORTRAN-77 STANDARD:
C                     1. KEYWORD "READONLY" IN OPEN STATEMENT
C                     2. EMBEDDED COMMENTS PREFACED WITH AN EXCLAMATION MARK (!) FOLLOWING VARIABLE DECLARATIONS
C                     3. ARGUMENT LIST BUILT-IN FUNCTION "%REF"
C                     4. CALLS TO VAX SYSTEM ROUTINES "SYS$BINTIM", "SYS$SCHDWK", "SYS$HIBER"  (SUBROUTINE DELAY)
C
C      OUTPUT:        GRAPHIC OUTPUT ONLY
C
C      AUTHORS:       PAUL REASENBERG AND DAVID OPPENHEIMER, U.S.G.S. IN MENLO PARK.  SOME OF THE ROUTINES
C                     WERE ADOPTED FROM CODE WRITTEN BY JOHN LAHR, BRUCE JULIAN, AND FRED KLEIN.
C
      REAL              AIN                               ! RAY ANGLE OF INCIDENCE
      REAL              AZM                               ! RAY AZIMUTH
      REAL              CX                                ! X POSITION OF STEREO NET CENTER
      REAL              CXMAX                             ! GREATEST X POSITION IN PLOT
      REAL              CY                                ! Y POSITION OF STEREO NET CENTER
      REAL              DA1                               ! DIP ANGLE OF PRINCIPLE PLANE
      REAL              DA2                               ! DIP ANGLE OF AUXILLIARY PLANE
      REAL              DD1                               ! DIP DIRECTION OF PRINCIPLE PLANE
      REAL              DD2                               ! DIP DIRECTION OF AUXILLIARY PLANE
      CHARACTER*1       DISC                              ! FLAG: IF "*" THEN FIRST MOTION DISCREPANT WITH SOLUTION
      REAL              DIST                              ! EPICENTRAL DISTANCE
      REAL              DX                                ! INCREMENTAL X POSITION BETWEEN BEACHBALLS
      REAL              DY                                ! INCREMENTAL Y POSITION BETWEEN BEACHBALLS
      CHARACTER*132     EVENT                             ! HYPO71 SUMMARY CARD
      CHARACTER*50      FILNAM                            ! FILE NAME OF DATA
      CHARACTER*1       HEAD                              ! HEADER FLAG: N=PLOT EVENT NUMBER, D=EVENT DATE
      REAL              HITE                              ! HEIGHT OF EVENT #
      INTEGER           I                                 ! LOOP INDEX OVER NUMBER OF LINES OF NEARBY SOLUTIONS
      INTEGER           ILINE                             ! INPUT LINE NUMBER
      INTEGER           IOS                               ! IO STATUS DESCRIPTOR
      CHARACTER*40      LINE                              ! SCRATCH VARIABLE FOR PLOT OUTPUT
      CHARACTER*1       MULT                              ! FLAG: Y(N)=DO (NOT) PLOT MULTIPLE SOLUTIONS
      INTEGER           MXEVNT                            ! MAXIMUM NUMBER OF EVENTS PER PAGE
      CHARACTER*4       NAME                              ! STATION NAME
      INTEGER           NCHAR                             ! NUMBER OF CHARACTERS TO BE PLOTTED
      INTEGER           NEV                               ! EVENT NUMBER
      INTEGER           NLABEL                            ! CURRENT EVENT LABEL NUMBER
      INTEGER           NLINE                             ! NUMBER OF LINES OF NEARBY SOLUTIONS
      INTEGER           NSKIP                             ! NUMBER OF EVENTS TO SKIP
      INTEGER           NSTAR                             ! NUMBER OF SOLUTIONS WITH FIT WITHIN  5% OF BEST SOLUTION
      INTEGER           NUM                               ! NUMBER OF EVENTS PLOTTED
      INTEGER           NY                                ! NUMBER OF BEACHBALLS IN Y DIRECTION
      REAL              PI                                ! PI
      REAL              PLOTLN                            ! PLOT LENGTH IN X DIRECTION
      CHARACTER*1       PLTPOL                            ! FLAG: Y(N)=DO (NOT) PLOT FIRST MOTION DATA
      CHARACTER*4       PRMK                              ! FIRST MOTION DESCRIPTION (EG. IPU0)
      REAL              RAD                               ! PI/180
      REAL              RMAX1                             ! RADIUS OF STEREO NET
      REAL              RMAX2                             ! RADIUS OF FIRST MOTION SYMBOL
      REAL              SA1                               ! RAKE OF AUXILLIARY PLANE
```

```
      REAL              SA2                         ! RAKE OF AUXILLIARY PLANE
      CHARACTER*1       STAR                        ! MULTIPLE INDICATOR
      CHARACTER*1       SYM                         ! FIRST MOTION DIRECTION
      CHARACTER*80      TITLE                       ! DATA SET DESCRIPTOR
      REAL              WT                          ! WEIGHT ASSIGNED TO PICK QUALITY IN PROGRAM FPFIT
      REAL              XPOS                        ! X PLOT POSITION
      REAL              YPOS                        ! Y PLOT POSITION
C
      PARAMETER (HITE = Ø.15, MXEVNT = 42, RMAX1 = Ø.5, RMAX2 = .Ø65,
     & YPAGE = 1Ø.5)
      PI = ATAN(1.Ø)*4.Ø
      RAD = PI/18Ø.Ø
      DY = RMAX1*3
      NY = YPAGE/DY
      DX = DY
C
1Ø    WRITE (6, 15) 'ENTER NAME OF DATA FILE:  '
15    FORMAT ('$', A)
      READ (5, 16, ERR = 1Ø) FILNAM
16    FORMAT (A)
      OPEN (UNIT = 2, FILE = FILNAM, ERR = 1Ø, STATUS = 'OLD', BLANK =
     & 'ZERO', READONLY)
      WRITE (6, 15) 'DO YOU WANT TO PLOT EVENT HEADERS AS NUMBERS (N) OR
     & DATES (D)?  '
2Ø    READ (5, 16, IOSTAT = IOS) HEAD
      IF (HEAD .EQ. 'y') HEAD = 'Y'
      IF (HEAD .EQ. 'n') HEAD = 'N'
      IF ((HEAD .NE. 'N' .AND. HEAD .NE. 'D') .OR. IOS .NE. Ø) THEN
        PRINT *, '***** PLEASE ANSWER "N" OR "D"; TRY AGAIN *****'
        GOTO 2Ø
      END IF
      WRITE (6, 15) 'ENTER NUMBER OF MECHANISMS TO SKIP (INCLUDING MULTI
     &PLE SOLUTIONS):  '
3Ø    READ (5, *, IOSTAT = IOS) NSKIP
      IF (NSKIP .LT. Ø .OR. IOS .NE. Ø) THEN
        PRINT *, '***** INVALID NUMBER; TRY AGAIN *****'
        GOTO 3Ø
      END IF
      WRITE (6,15) 'DO YOU WANT TO PLOT MULTIPLE SOLUTIONS (Y OR N)?  '
4Ø    READ (5, 16, IOSTAT = IOS) MULT
      IF (MULT .EQ. 'y') MULT = 'Y'
      IF (MULT .EQ. 'n') MULT = 'N'
      IF ((MULT .NE. 'Y' .AND. MULT .NE. 'N') .OR. IOS .NE. Ø) THEN
        PRINT *, '***** PLEASE ANSWER "Y" OR "N"; TRY AGAIN *****'
        GOTO 4Ø
      END IF
      WRITE (6,15) 'DO YOU WANT TO PLOT FIRST MOTION DATA (Y OR N)?  '
5Ø    READ (5, 16, IOSTAT = IOS) PLTPOL
      IF (PLTPOL .EQ. 'y') PLTPOL = 'Y'
      IF (PLTPOL .EQ. 'n') PLTPOL = 'N'
      IF ((PLTPOL .NE. 'Y' .AND. PLTPOL .NE. 'N') .OR. IOS .NE. Ø) THEN
        PRINT *, '***** PLEASE ANSWER "Y" OR "N"; TRY AGAIN *****'
        GOTO 5Ø
      END IF
C
C READ HYPO71 HEADER CARD (FIRST LINE IN MODEL FILE)
C
      NLABEL = Ø
      NEV = Ø
      NUM = Ø
      CXMAX = Ø.
      ILINE = 1
      READ (2, 6Ø, ERR = 2ØØØ) TITLE
6Ø    FORMAT (A)
```

```
C
C INITIALIZE PLOT PROGRAM
C
      CALL PLOTS (Ø., Ø., Ø)
      CALL ERASE
      CALL DELAY ('ØØØØ ØØ:ØØ:Ø1.ØØ')
      CALL PLOT (.1, .1, -3)
      CALL NEWPEN (2)
C
C READ EVENT
C
7Ø      ILINE = ILINE + 1
        READ (2, 8Ø, END = 1ØØØ, ERR = 2ØØØ) EVENT
8Ø      FORMAT (1X, A132)
        NEV = NEV + 1
        READ (EVENT, 9Ø, ERR = 2ØØØ) DD1, DA1, SA1, STAR
9Ø      FORMAT (T81, F4.Ø, F3.Ø, F4.Ø, T132, A1)
        IF (STAR .EQ. ' ') NLABEL = NLABEL + 1
        IF ((NEV .GT. NSKIP) .AND. ((MULT .EQ. 'N' .AND. STAR .EQ. ' ')
     & .OR. MULT .EQ. 'Y')) THEN
C
C END PLOT IF MORE THAN MXEVNT EVENTS
C
          NUM = NUM + 1
          IF (NUM .GT. MXEVNT) THEN
            NUM = 1
            CALL PLOT (Ø., Ø., -999)
            CALL ERASE
            CALL DELAY ('ØØØØ ØØ:ØØ:Ø1.ØØ')
            CALL PLOT (.1, .1, -3)
            CALL NEWPEN (2)
          END IF
          CY = YPAGE - DY*FLOAT(JMOD(NUM - 1, NY)) - RMAX1*2.Ø
          CX = RMAX1 + DX*FLOAT((NUM - 1)/NY )
          IF (JMOD(NUM - 1, NY) .EQ. Ø) CXMAX = CXMAX + DX
          XPOS = CX - RMAX1
          YPOS = CY + RMAX1 + .1
          IF (HEAD .EQ. 'N') THEN
            IF (NLABEL .LT. 1Ø) THEN
              NCHAR = 2
              WRITE (LINE, '(I1, A1)') NLABEL, STAR
            ELSE IF (NLABEL .GE. 1Ø .AND. NLABEL .LT. 1ØØ) THEN
              NCHAR = 3
              WRITE (LINE, '(I2, A1)') NLABEL, STAR
            ELSE
              NCHAR = 4
              WRITE (LINE, '(I3, A1)') NLABEL, STAR
            END IF
            CALL SYMBOL (XPOS, YPOS, HITE, %REF(LINE), Ø., NCHAR)
          ELSE
            WRITE (LINE, '(A11, A1)') EVENT(1:11), STAR
            NCHAR = 12
            CALL SYMBOL (XPOS, YPOS, HITE*.75, %REF(LINE), Ø., NCHAR)
          END IF
C
C PLOT STEREO NET PERIMETERS
C
          CALL STRNT1 (CX, CY, RAD, RMAX1)
C
C PLOT NODAL PLANES AND "P" AND "T" AXES
C
          CALL PLOTPL (CX, CY, DA1, PI, RAD, RMAX1, DD1 - 9Ø.)
          CALL AUXPLN (DD1, DA1, SA1, DD2, DA2, SA2)
          CALL PLOTPL (CX, CY, DA2, PI, RAD, RMAX1, DD2 - 9Ø.)
```

```
            CALL TPPLOT (CX, CY, DA1, DD1, RMAX2*1.5, PI, PLTPOL, RAD,
     & RMAX1, SA1, 1.Ø)
          END IF
C
C SKIP OVER NEARBY SOLUTIONS
C
          ILINE = ILINE + 1
          READ (2, *, ERR = 2ØØØ) NSTAR
          NLINE = (NSTAR-1)/11 + 1
          DO 11Ø I = 1, NLINE
            ILINE = ILINE + 1
            READ (2, 1ØØ, ERR = 2ØØØ) LINE
1ØØ         FORMAT (A)
11Ø       CONTINUE
C
C READ PHASE CARD
C
12Ø       ILINE = ILINE + 1
          READ (2, 13Ø, END = 1ØØØ, ERR = 2ØØØ) NAME, DIST, AZM, AIN,
     & PRMK, WT, DISC
13Ø       FORMAT (1X, A4, 3F6.1, 3X, A4, F5.2, 2X, A1)
          IF (NAME .NE. '    ') THEN
            IF((NEV .GT. NSKIP) .AND. ((MULT .EQ. 'N' .AND. STAR .EQ. ' ')
     & .OR. MULT .EQ. 'Y') .AND. (PLTPOL .EQ. 'Y')) THEN
C
C PLOT FIRST MOTION
C
              IF (PRMK(3:3) .EQ. 'U' .OR. PRMK(3:3) .EQ. '+') THEN
                SYM = 'C'
              ELSE
                SYM = 'D'
              END IF
              CALL PLTSYM (AIN, AZM, CX, CY, RMAX2, '    ', PI, RAD,
     & RMAX1, SYM, 1.Ø)
            END IF
            GOTO 12Ø
          END IF
        GOTO 7Ø
C
C END OF FILE
C
1ØØØ    CALL PLOT (Ø., Ø., 999)
        PLOTLN = CXMAX + RMAX1
        PRINT *, 'MAXIMUM PLOT SIZE =', PLOTLN, 'X', YPAGE
        CLOSE (2)
        STOP
C
C  READ ERROR
C
2ØØØ    PRINT *, ' READ ERROR ON LINE', ILINE
        STOP
        END
C
C
C
C
C
C
        SUBROUTINE AUXPLN (DD1, DA1, SA1, DD2, DA2, SA2)
C
C     CALCULATE THE AUXILLIARY PLANE OF A DOUBLE COUPLE FAULT PLANE SOLUTION, GIVEN THE PRINCIPLE PLANE.
C
C     WRITTEN BY PAUL REASENBERG, JUNE, 1984, FROM CLASS NOTES BY DAVE BOORE, (BOTH AT THE U.S.G.S., MENLO PARK.)
C     ANGLE VARIABLES PHI, DEL AND LAM ARE AS DEFINED IN AKI AND RICHARDS, (198Ø), P.114.
```

```
C
      REAL              DA1                              ! DIP ANGLE IN DEGREES
      REAL              DD1                              ! DIP DIRECTIONS IN DEGREES
      REAL              SA1                              ! SLIP ANGLE IN DEGREES
      REAL              DA2                              ! DIP ANGLE OF AUXILLIARY PLANE
      REAL              DD2                              ! DIP DIRECTION OF AUXILLIARY PLANE
      REAL              SA2                              ! SLIP ANGLE OF AUXILLARY PLANE
C
      DOUBLE PRECISION  BOT                              ! SCRATCH VARIABLE
      DOUBLE PRECISION  DEL1                             ! DIP ANGLE OF PRINCIPAL PLANE IN RADIANS
      LOGICAL           FIRST                            ! TEST: TRUE IF FIRST TIME INTO ROUTINE
      DOUBLE PRECISION  PHI1                             ! FAULT PLANE STRIKE OF PRINCIPAL PLANE
      DOUBLE PRECISION  PHI2                             ! STRIKE OF AUXILLIARY PLANE IN RADIANS
      DOUBLE PRECISION  RAD                              ! CONVERSION FACTOR FROM DEGREES TO RADIAN
      DOUBLE PRECISION  SGN                              ! SAVES PRINCIPAL PLANE SLIP ANGLE FOR ASSIGNING PROPER SIGN TO AUXILLIARY
      DOUBLE PRECISION  TOP                              ! SCRATCH VARIABLE
      DOUBLE PRECISION  XLAM1                            ! SLIP ANGLE OF PRINCIPAL PLANE IN RADIANS
      DOUBLE PRECISION  XLAM2                            ! SLIP ANGLE OF AUXILLIARY PLANE
C
      DATA FIRST /.TRUE./
      SAVE FIRST, RAD
C
      IF (FIRST) THEN
        FIRST = .FALSE.
        RAD = DATAN(1.ØDØ)/45.ØDØ
      END IF
C
      PHI1 = DD1 - 9Ø.ØDØ
      IF (PHI1 .LT. Ø.ØDØ) PHI1 = PHI1 + 36Ø.ØDØ
      PHI1 = PHI1*RAD
      DEL1 = DA1*RAD
      SGN = SA1
      XLAM1 = SA1*RAD
C
      TOP = DCOS(XLAM1)*DSIN(PHI1) - DCOS(DEL1)*DSIN(XLAM1)*DCOS(PHI1)
      BOT = DCOS(XLAM1)*DCOS(PHI1) + DCOS(DEL1)*DSIN(XLAM1)*DSIN(PHI1)
      DD2 = DATAN2(TOP, BOT)/RAD
      PHI2 = (DD2 - 9Ø.ØDØ)*RAD
      IF (SA1 .LT. Ø.ØDØ) DD2 = DD2 - 18Ø.ØDØ
      IF (DD2 .LT. Ø.ØDØ) DD2 = DD2 + 36Ø.ØDØ
      IF (DD2. GT. 36Ø.ØDØ) DD2 = DD2 - 36Ø.ØDØ
C
      DA2 = DACOS(DSIN(DABS(XLAM1))*DSIN(DEL1))/RAD
      XLAM2 = -DCOS(PHI2)*DSIN(DEL1)*DSIN(PHI1) +
     & DSIN(PHI2)*DSIN(DEL1)*DCOS(PHI1)
C
C MACHINE ACCURACY PROBLEM
C
      IF (XLAM2 .GT. 1.ØDØ) THEN
        XLAM2 = 1.ØDØ
      END IF
      XLAM2 = DSIGN(DACOS(XLAM2), SGN)
      SA2 = XLAM2/RAD
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE CIRCLE (SIZE, TWOPI, XØ, YØ)
C
```

```
C PLOT A CIRCLE
C
      REAL              SIZE                        ! CIRCLE DIAMETER
      REAL              TWOPI                       ! TWO*PI
      REAL              XØ                          ! X POSTION OF CENTER
      REAL              YØ                          ! Y POSTION OF CENTER
C
      REAL              ANGLE                       ! ANGLE
      INTEGER           J                           ! LOOP INDEX
      INTEGER           N                           ! NUMBER OF POINTS OF WHICH CIRCLE COMPOSED
      REAL              SIZE2                       ! CIRCLE RADIUS
      REAL              X                           ! X PLOT POSTION
      REAL              Y                           ! Y PLOT POSTION
C
      SIZE2 = SIZE*Ø.5
C
C COMPUTE OPTIMUM # OF POINTS TO DRAW
C
      N = 2Ø*SQRT(SIZE2*2Ø.)
      IF (N .LT. 1Ø) N = 1Ø
C
C DRAW CIRCLE
C
      X = XØ + SIZE2
      CALL PLOT (X, YØ, 3)
      DO 1Ø J = 1, N
        ANGLE = TWOPI*FLOAT(J)/FLOAT(N)
        X = XØ + SIZE2*COS(ANGLE)
        Y = YØ + SIZE2*SIN(ANGLE)
        CALL PLOT (X, Y, 2)
1Ø    CONTINUE
C
      RETURN
      END
C
C
C
C
C
        SUBROUTINE DELAY (TIME)
C
C CREATES A DELAY FOR SCREEN RECOVERY
C
      CHARACTER*(*)     TIME                        ! STANDARD VMS TIME FORMAT OF THE FORM 'DDDD HH:MM:SS.SS'
      DOUBLE PRECISION  B                           ! BINARY TIME CORRESPONDING TO TIME
C
C CONVERT ASCII TIME TO BINARY TIME
C
      CALL SYS$BINTIM (TIME,B)
C
C SCHEDULE A WAKEUP FOR A ELAPSED TIME (NEGATIVE B)
C
      CALL SYS$SCHDWK (,,-B,)
C
C HIBERNATE AND REAWAKE
C
      CALL SYS$HIBER
C
      RETURN
      END
C
C
C
C
```

```
C
      SUBROUTINE ERASE
C
C SENDS A SCREEN ERASE CODE TO A TEKTRONIX TERMINAL
C
      CHARACTER*1         A                               ! ASCII ESCAPE
      CHARACTER*1         B                               ! ASCII FORM FEED
C
      A = CHAR(27)
      B = CHAR(12)
      WRITE (6, 10) A, B
10    FORMAT (1X, 2A1)
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE PLOTPL (CX, CY, DPIDG, PI, RAD, RMAX, STRKDG)
C
C PLOTS FAULT PLANE ON LOWER HEMISPHERE STEREO NET
C
      REAL                CX                              ! X POSITION OF CIRCLE CENTER
      REAL                CY                              ! Y POSITION OF CIRCLE CENTER
      REAL                DPIDG                           ! DIP ANGLE IN DEGREES
      REAL                PI                              ! PI
      REAL                RAD                             ! PI/180
      REAL                RMAX                            ! RADIUS OF CIRCLE
      REAL                STRKDG                          ! STRIKE ANGLE IN DEGREES

      REAL                ANG                             ! ANGLE IN RADIANS
      REAL                AINP(91)                        ! ANGLE OF INCIDENCE IN RADIANS
      REAL                ARG                             ! DUMMY ARGUMENT
      REAL                AZ                              ! AZIMUTH
      REAL                CON                             ! RADIUS COEFFICIENT
      REAL                DIPRD                           ! DIP ANGLE IN RADIANS
      INTEGER             I                               ! LOOP INDEX
      INTEGER             MI                              ! SCRATCH INDEX
      REAL                RADIUS                          ! RADIUS
      REAL                SAZ(91)                         ! AZIMUTH IN RADIANS
      REAL                STRKRD                          ! STRIKE IN RADIANS
      REAL                TAZ                             ! SCRATCH VARIABLE
      REAL                TPD                             ! SCRATCH VARIABLE
      REAL                X                               ! X PLOT POSITION
      REAL                Y                               ! Y PLOT POSITION
C
      STRKRD = STRKDG*RAD
      DIPRD = DPIDG*RAD
      TPD = TAN(PI*.5 - DIPRD)**2
C
C CASE OF VERTICAL PLANE
C
      IF (DPIDG .EQ. 90.0) THEN
        X = RMAX*SIN(STRKRD) + CX
        Y = RMAX*COS(STRKRD) + CY
        CALL PLOT (X, Y, 3)
        X = RMAX*SIN(STRKRD + PI) + CX
        Y = RMAX*COS(STRKRD + PI) + CY
        CALL PLOT (X, Y, 2)
        RETURN
      END IF
C
```

```
C COMPUTE ANGLE OF INCIDENCE, AZIMUTH
C
      DO 1Ø I = 1, 9Ø
        ANG = FLOAT(I - 1)*RAD
        ARG = SQRT((COS(DIPRD)**2)*(SIN(ANG)**2))/COS(ANG)
        SAZ(I) = ATAN(ARG)
        TAZ = TAN(SAZ(I))**2
        ARG = SQRT(TPD + TPD*TAZ + TAZ)
        AINP(I) = ACOS(TAN(SAZ(I))/ARG)
  1Ø  CONTINUE
      SAZ(91) = 9Ø.*RAD
      AINP(91) = PI*.5 - DIPRD
C
C PLOT PLANE
C
      CON = RMAX*SQRT(2.)
      DO 2Ø I = 1, 18Ø
        IF (I .LE. 91) THEN
          MI = I
          AZ = SAZ(I) + STRKRD
        ELSE
          MI = 181 - I
          AZ = PI - SAZ(MI) + STRKRD
        END IF
        RADIUS = CON*SIN(AINP(MI)*Ø.5)
        X = RADIUS*SIN(AZ) + CX
        Y = RADIUS*COS(AZ) + CY
        IF (I .EQ. 1) THEN
          CALL PLOT (X, Y, 3)
        ELSE
          CALL PLOT (X, Y, 2)
        END IF
2Ø    CONTINUE
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE STRNT1 (CX, CY, RAD, RMAX)
C
C PLOT PERIMETER OF A STEREO NET
C
      REAL             CX                            ! X POSITION OF CIRCLE CENTER
      REAL             CY                            ! Y POSITION OF CIRCLE CENTER
      REAL             RAD                           ! PI/18Ø
      REAL             RMAX                          ! RADIUS OF CIRCLE

      REAL             CSIZ                          ! SCRATCH VARIABLE (RMAX/1ØØ)
      INTEGER          I                             ! LOOP INDEX OVER DEGREES
      INTEGER          N                             ! TESTS 1Ø DEGREE TICK POSITION
      INTEGER          NN                            ! TESTS 9Ø DEGREE TICK POSITION
      REAL             P                             ! TICK LENGTH
      REAL             PHI                           ! AZIMUTH IN RADIANS
      REAL             X                             ! X POSTION OF CIRCLE
      REAL             XP                            ! X POSITION OF END OF TICK
      REAL             Y                             ! Y POSTION OF CIRCLE
      REAL             YP                            ! Y POSITION OF END OF TICK
C
      CALL NEWPEN (2)
C
C DRAW CIRCLE @ 5 DEGREE INCREMENTS
```

```
C
      DO 10 I = 1, 73
        PHI = FLOAT(I - 1)*RAD*5.0
        X = RMAX*COS(PHI) + CX
        Y = RMAX*SIN(PHI) + CY
        N = (I - 1) - ((I - 1)/10)*10
        NN = (I - 1) - ((I - 1)/90)*90
        IF ((N .EQ. 0) .AND. (I .GT. 10)) THEN
          P = 0.02*RMAX
        ELSE IF ((NN .EQ. 0) .AND. (I .GT. 90)) THEN
          P = 0.04*RMAX
        ELSE
          P = 0.01*RMAX
        END IF
        XP = (RMAX + P)*COS(PHI) + CX
        YP = (RMAX + P)*SIN(PHI) + CY
        IF (I .GT. 1) THEN
          CALL PLOT (X, Y, 2)
          IF (JMOD(I - 1, 18) .EQ. 0) THEN
            XP = (RMAX + TIC)*COS(PHI) + CX
            YP = (RMAX + TIC)*SIN(PHI) + CY
            CALL PLOT (XP, YP, 2)
            CALL PLOT (X, Y, 3)
         END IF
        ELSE
          CALL PLOT (X, Y, 3)
        END IF
10    CONTINUE
C
C PLOT + AT CENTER
C
      CSIZ = .01*RMAX
      CALL PLOT (CX - CSIZ, CY, 3)
      CALL PLOT (CX + CSIZ, CY, 2)
      CALL PLOT (CX, CY - CSIZ, 3)
      CALL PLOT (CX, CY + CSIZ, 2)
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE TPPLOT (CX, CY, DAI, DDI, HITE, PI, PLTPOL, RAD, RMAX,
     & SAI, WT)
C
C PLOT P AND T AXES
C
      REAL              CX                           ! X POSITION OF CIRCLE CENTER
      REAL              CY                           ! Y POSITION OF CIRCLE CENTER
      REAL              DAI                          ! DIP ANGLE
      REAL              DDI                          ! DIP DIRECTION
      REAL              HITE                         ! HEIGHT OF P,T SYMBOL
      REAL              PI                           ! PI
      CHARACTER*1       PLTPOL                       ! FLAG: Y(N)=DO (NOT) PLOT FIRST MOTION DATA
      REAL              RAD                          ! PI/180
      REAL              RMAX                         ! RADIUS OF CIRCLE
      REAL              SAI                          ! RAKE
      REAL              WT                           ! WEIGHT ASSIGNED TO PICK QUALITY IN PROGRAM FPFIT
C
      REAL              AIN1                         ! ANGLE OF INCIDENCE OF P/T AXIS
      REAL              AIN2                         ! ANGLE OF INCIDENCE OF T/P AXIS
      REAL              ANG                          ! ANGLE OF PLOT SYMBOL
```

```
      REAL              AZ1                                  ! AZIMUTH OF P/T AXIS
      REAL              AZ2                                  ! AZIMUTH OF T/P AXIS
      CHARACTER*4       BLANK                                ! BLANK
      REAL              DA2                                  ! DIP ANGLE OF AUXILIARY PLANE
      REAL              DD2                                  ! DIP DIRECTION OF AUXILIARY PLANE
      REAL              SA2                                  ! STRIKE OF AUXILIARY PLANE
      CHARACTER*1       SYM1                                 ! P/T PLOT SYMBOL
      CHARACTER*1       SYM2                                 ! T/P PLOT SYMBOL
C
      PARAMETER (ANG = Ø.Ø, BLANK = '    ')
C
C FIND AUXILLIARY PLANE
C
      CALL AUXPLN (DD1, DA1, SA1, DD2, DA2, SA2)
C
C FIND P AND T AXES
C
      CALL TANDP (AIN1, AIN2, AZ1, AZ2, DA1, DA2, DD1, DD2, PI, RAD)
      IF (SA1 .LT. Ø.) THEN
        SYM1 = 'P'
        SYM2 = 'T'
      ELSE
        SYM1 = 'T'
        SYM2 = 'P'
      END IF
C
C PLOT SYMBOLS
C
      IF (SYM1 .EQ. 'T' .OR. PLTPOL .EQ. 'Y') CALL PLTSYM (AIN1, AZ1,
     & CX, CY, HITE, BLANK, PI, RAD, RMAX, SYM1, WT)
      IF (SYM2 .EQ. 'T' .OR. PLTPOL .EQ. 'Y') CALL PLTSYM (AIN2, AZ2,
     & CX, CY, HITE, BLANK, PI, RAD, RMAX, SYM2, WT)
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE TANDP(AIN1, AIN2, AZ1, AZ2, DA1, DA2, DD1, DD2, PI,RAD)
C
C     GIVEN TWO PLANES COMPUTE AZ AND ANGLE OF INCIDENCE OF P & T AXES
C
      REAL              AIN1                                 ! ANGLE OF INCIDENCE OF P/T AXIS
      REAL              AIN2                                 ! ANGLE OF INCIDENCE OF T/P AXIS
      REAL              AZ1                                  ! AZIMUTH OF P/T AXIS
      REAL              AZ2                                  ! AZIMUTH OF T/P AXIS
      REAL              DA1                                  ! DIP ANGLE OF PRINICIPLE PLANE
      REAL              DA2                                  ! DIP ANGLE OF AUXILLIARY PLANE
      REAL              DD1                                  ! DIP DIRECTION OF PRINCIPLE PLANE
      REAL              DD2                                  ! DIP DIRECTION OF AUXILLIARY PLANE
      REAL              PI                                   ! PI
      REAL              RAD                                  ! PI/18Ø
C
      REAL              ALAT1                                ! DIP ANGLE IN RADIANS OF PRINCIPLE PLANE MEASURED FROM VERTICAL
      REAL              ALAT2                                ! DIP ANGLE IN RADIANS OF AUXILLIARY PLANE MEASURED FROM VERTICAL
      REAL              ALON1                                ! DD1 IN RADIANS
      REAL              ALON2                                ! DD2 IN RADIANS
      REAL              AZIMTH                               ! AZIMUTH IN RADIANS OF POLE ??
      REAL              AZØ                                  ! AZIMUTH FROM POLE OF AUXILLIARY PLANE TO POLE OF PRINCIPLE ??
      REAL              BAZM                                 ! NOT USED
      REAL              DELTA                                ! NOT USED
      REAL              PLUNGE                               ! PLUNGE IN RADIANS OF POLE ??
```

```
      REAL              SHIFT                      ! AZIMUTHAL SHIFT FROM POLE OF PLANE TO P TO T AXIS (= 45 DEGREES)??
      REAL              XPOS                       ! NOT USED
      REAL              YPOS                       ! NOT USED
C
      PARAMETER (SHIFT = Ø.7853981)
C
      ALAT1 = (9Ø. - DA1)*RAD
      ALON1 = DD1*RAD
      ALAT2 = (9Ø. - DA2)*RAD
      ALON2 = DD2*RAD
      CALL REFPT (ALAT2, ALON2)
      CALL DELAZ (ALAT1, ALON1, DELTA, AZØ, BAZM, XPOS, YPOS)
      CALL BACK (SHIFT, AZØ, PLUNGE, AZIMTH)
      IF (ABS(AZIMTH) .GT. PI) AZIMTH = AZIMTH - SIGN(2.Ø*PI, AZIMTH)
      AZ1 = AZIMTH/RAD
      AIN1 = PLUNGE/RAD + 9Ø.
      AZØ = AZØ + PI
      CALL BACK (SHIFT, AZØ, PLUNGE, AZIMTH)
      IF (ABS(AZIMTH) .GT. PI) AZIMTH = AZIMTH - SIGN(2.Ø*PI, AZIMTH)
      AZ2 = AZIMTH/RAD
      AIN2 = PLUNGE/RAD + 9Ø.
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE PLTSYM (AIN, AZ, CX, CY, HITE, NAME, PI, RAD, RMAX,
     & SYM, WT)
C
C    PLOT EITHER FIRST MOTION SYMBOL (C,D,+,-) WITH STATION NAME NEXT TO SYMBOL, OR STRESS AXES SYMBOL (P & T)
C
      REAL              AIN                        ! ANGLE OF INCIDENCE OF SYMBOL
      REAL              AZ                         ! AZIMUTH OF SYMBOL
      REAL              CX                         ! X POSITION OF CIRCLE CENTER
      REAL              CY                         ! Y POSITION OF CIRCLE CENTER
      REAL              HITE                       ! HEIGHT OF SYMBOL
      CHARACTER*4       NAME                       ! STRING TO BE PLOTTED TO RIGHT OF SYMBOL
      REAL              PI                         ! PI
      REAL              RAD                        ! PI/18Ø
      REAL              RMAX                       ! RADIUS OF CIRCLE
      CHARACTER*1       SYM                        ! PLOT SYMBOL
      REAL              WT                         ! WEIGHT ASSIGNED TO PICK QUALITY IN PROGRAM FPFIT

      REAL              AINR                       ! AIN IN RADIANS
      REAL              ANG                        ! ANGLE OF CIRCLE BISECTOR FOR COMPRESSIONAL FILL
      REAL              ANGLE                      ! SAME AS ANG
      REAL              AZR                        ! AZ IN RADIANS
      REAL              CON                        ! RMAX * SQRT(2.Ø)
      REAL              DANG                       ! ANGLE INCREMENT OF CIRCLE BISECTOR FOR COMPRESSIONAL FILL
      LOGICAL           FIRST                      ! FLAG FIRST TIME INTO ROUTINE
      REAL              R                          ! DISTANCE FROM CX, CY TO PLOT POSITION
      REAL              X                          ! X POSITION OF SYMBOL
      REAL              XPOS                       ! X POSITION OF CIRCLE BISECTOR FOR COMPRESSIONAL FILL
      REAL              Y                          ! Y POSITION OF SYMBOL
      REAL              YPOS                       ! Y POSITION OF CIRCLE BISECTOR FOR COMPRESSIONAL FILL
C
      AZR = AZ*RAD
      AINR = AIN*RAD
C
C UPGOING RAYS
C
```

```
      IF (AIN .GT. 9Ø.) THEN
        AINR = PI - AINR
        AZR = PI + AZR
      END IF
      CON = RMAX*SQRT(2.Ø)
      R = CON*SIN(AINR*Ø.5)
      X = R*SIN(AZR) + CX
      Y = R*COS(AZR) + CY
C
C STRESS AXIS SYMBOL
C
      IF ((SYM .EQ. 'P') .OR. (SYM .EQ. 'T')) THEN
        X = X - .286*HITE
        Y = Y - .5*HITE
        CALL SYMBOL (X, Y, HITE, %REF(SYM), Ø.Ø, 1)
      ELSE
C
C FIRST MOTION SYMBOL
C
        CALL CIRCLE (HITE, 2.Ø*PI, X, Y)
C
C  FILL IN COMPRESSION CIRCLE
C
        IF (SYM .EQ. 'C') THEN
          N = 9
          ANG = Ø.
          DANG = PI/FLOAT(N)
          DO 1Ø I = 1, N
            ANG = ANG + DANG
            ANGLE = ANG
            XPOS = X + .5*HITE*COS(ANGLE)
            YPOS = Y + .5*HITE*SIN(ANGLE)
            CALL PLOT (XPOS, YPOS, 3)
            ANGLE = ANG + PI
            XPOS = X + .5*HITE*COS(ANGLE)
            YPOS = Y + .5*HITE*SIN(ANGLE)
            CALL PLOT (XPOS, YPOS, 2)
1Ø        CONTINUE
        END IF
C
C PLOT STATION NAME
C
        IF (NAME .NE. '    ') CALL SYMBOL (X + HITE/2., Y, HITE/2.,
     & %REF(NAME), Ø., 4)
      END IF
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE GEOCEN
C
C      GEOCEN - CALCULATE GEOCENTRIC POSTITIONS, DISTANCES, AND AZIMUTHS (BRUCE JULIAN, USGS MENLO PARK, CA)
C
C      THE GEOCENTRIC DISTANCE DELTA AND AZIMUTH AZØ FROM POINT (LATØ, LONØ) TO POINT (LAT1, LON1) ARE CALCULTED FROM
C            COS(DELTA) = COS(LATØ')*COS(LAT1')*COS(LON1 - LONØ) + SIN(LATØ')*SIN(LAT1')
C            SIN(DELTA) = SQRT(A*A + B*B)
C            TAN(AZØ) = A/B
C
C      WHERE
C            A = COS(LAT1')*SIN(LON1-LONØ)
```

```
C           B = COS(LATNØ')*SIN(LAT1') - SIN(LATØ')*COS(LAT1')*COS(LON1 - LONØ)
C           LATØ', LAT1' = GEOCENTRIC LATITUDES OF POINTS
C           LONØ, LON1 = LONGITUDES OF POINTS
C
C      THE GEOCENTRIC LATITUDE LAT' IS GOTTEN FROM THE GEOGRAPHIC LATITUDE LAT BY TAN(LAT') = (1 - ALPHA)*(1 - ALPHA)*TAN(LAT),
C      WHERE ALPHA IS THE FLATTENING OF THE ELLIPSOID.  SEE FUNCTION GGTOGC FOR CONVERSION.
C      THE BACK AZIMUTH IS CALCULATED BY THE SAME FORMULAS WITH (LATØ', LONØ) AND (LAT1', LON1) INTERCHANGED.
C      AZIMUTH IS MEASURED CLOCKWISE FROM NORTH THRU EAST.
C
      REAL              R, THETA                          ! RADIUS, AZIMUTH IN POLAR COORDINATES
      REAL              AZØ                               ! AZIMUTH FROM REFERENCE POINT TO SECONDARY POINT IN RADIANS
      REAL              AZ1                               ! AZIMUTH FROM SECONDARY POINT TO REFERENCE POINT IN RADIANS
      REAL              CDELT                             ! SINE OF DELTA TO SECONDARY POINT
      REAL              CDLON                             ! COSINE OF DIFFERENCE OF SECONDARY POINT, REFERENCE LONGITUDE
      REAL              COLAT                             ! AVERAGE COLATITUDE OF STATION
      REAL              CTØ                               ! SINE OF REFERENCE POINT LATITUDE
      REAL              CT1                               ! SINE OF SECONDARY POINT LATITUDE
      REAL              CZØ                               ! COSINE OF AZIMUTH TO SECONDARY POINT
      REAL              DELTA                             ! GEOCENTRIC DISTANCE IN DEGREES
      REAL              DLON                              ! AZIMUTH IN POLAR COORDINATES TO SECONDARY POINT ?
      REAL              ERAD                              ! EQUATORIAL RADIUS (CHOVITZ, 1981, EOS, 62, 65-67)
      REAL              FLAT                              ! EARTH FLATTENING CONSTANT (CHOVITZ, 1981, EOS, 62, 65-67)
      INTEGER           LAMBDA                            ! DUMMY VARIABLE
      REAL              LAT                               ! LATITUDE IN RADIANS
      REAL              LON                               ! LONGITUDE IN RADIANS
      REAL              OLAT                              ! ORIGIN LATITUDE IN RADIANS
      REAL              OLON                              ! ORIGIN LONGITUDE IN RADIANS
      REAL              PHIØ                              ! REFERENCE SECONDARY POINT LONGITUDE
      REAL              PI                                ! 3.14159...
      REAL              RADIUS                            ! EARTH RADIUS AT COLAT
      REAL              SDELT                             ! COSINE OF DELTA TO SECONDARY POINT
      REAL              SDLON                             ! SINE OF DIFFERENCE OF SECONDARY POINT, REFERENCE LONGITUDE
      REAL              STØ                               ! COSINE OF REFERENCE POINT LATITUDE
      REAL              ST1                               ! COSINE OF SECONDARY POINT LATITUDE
      REAL              TWOPI                             ! 2*PI
C
      SAVE STØ, CTØ, PHIØ, OLAT
      PARAMETER (PI = 3.14159265358979, TWOPI = 2.*PI)
      PARAMETER (FLAT = 1./298.257, ERAD = 6378.137)
      PARAMETER (LAMBDA = FLAT*(2. - FLAT)/(1. - FLAT)**2)
C
C REFPT - STORE THE GEOCENTRIC COORDINATES OF THE REFEERNCE POINT
C
      ENTRY REFPT(OLAT, OLON)
C
      STØ = COS(OLAT)
      CTØ = SIN(OLAT)
      PHIØ = OLON
      RETURN
C
C DELAZ - CALCULATE THE GEOCENTRIC DISTANCE, AZIMUTHS
C
      ENTRY DELAZ(LAT, LON, DELTA, AZØ, AZ1, X, Y)
C
      CT1 = SIN(LAT)
      ST1 = COS(LAT)
      IF ((CT1 - CTØ) .EQ. Ø. .AND. (LON - PHIØ) .EQ. Ø.) THEN
        DELTA = Ø.
        AZØ = Ø.
        AZ1 = Ø.
      ELSE
        SDLON = SIN(LON - PHIØ)
        CDLON = COS(LON - PHIØ)
        CDELT = STØ*ST1*CDLON + CTØ*CT1
```

```
         CALL CVRTOP (STØ*CTI - STI*CTØ*CDLON, STI*SDLON, SDELT, AZØ)
         DELTA = ATAN2(SDELT, CDELT)
         CALL CVRTOP (STI*CTØ - STØ*CTI*CDLON, -SDLON*STØ, SDELT, AZ1)
         IF (AZØ .LT. Ø.Ø) AZØ = AZØ + TWOPI
         IF (AZ1 .LT. Ø.Ø) AZ1 = AZ1 + TWOPI
      END IF
      COLAT = PI/2. - (LAT + OLAT)/2.
      RADIUS = ERAD/SQRT(1.Ø + LAMBDA*COS(COLAT)**2)
      Y = RADIUS*DELTA*COS(AZØ)
      X = RADIUS*DELTA*SIN(AZØ)
      RETURN
C
C BACK - CALCULATE GEOCENTRIC COORDINATES OF SECONDARY POINT FROM DELTA, AZ
C
      ENTRY BACK (DELTA, AZØ, LAT, LON)
C
      SDELT = SIN(DELTA)
      CDELT = COS(DELTA)
      CZØ = COS(AZØ)
      CTI = STØ*SDELT*CZØ + CTØ*CDELT
      CALL CVRTOP (STØ*CDELT - CTØ*SDELT*CZØ, SDELT*SIN(AZØ), STI, DLON)
      LAT = ATAN2(CTI, STI)
      LON = PHIØ + DLON
      IF (ABS(LON) .GT. PI) LON = LON - SIGN(TWOPI, LON)
C
      RETURN
      END
C
C
C
C
C
      SUBROUTINE CVRTOP(X, Y, R, THETA)
C
C CVRTOP - CONVERT FROM RECTANGULAR TO POLAR COORDINATES (BRUCE JULIAN, USGS MENLO PARK, CA)
C
      REAL              X,Y                                ! X,Y RECTANGULAR COORDINATES
      REAL              R, THETA                           ! RADIUS, AZIMUTH IN POLAR COORDINATES
C
      R = SQRT(X*X + Y*Y)
      THETA = ATAN2(Y, X)
      RETURN
      END
```